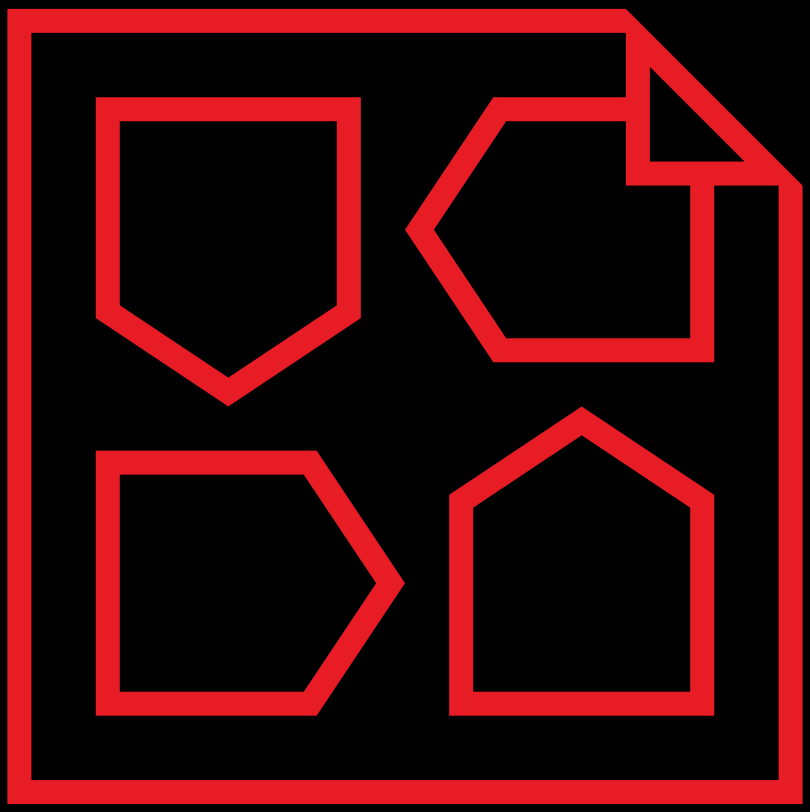
VFQ Value Flow Quality



Building plans

by emergn

This publication forms part of Value, Flow, Quality® Education. Details of this and other Emergn courses can be obtained from Emergn, 20 Harcourt Street, Dublin, D02 H364, Ireland or Emergn, 190 High St, Floor 4, Boston, MA 02110, USA.

Alternatively, you may visit the Emergn website at http://www.emergn.com/education where you can learn more about the range of courses on offer.

To purchase Emergn's Value, Flow, Quality® courseware visit http://www.valueflowquality.com, or contact us for a brochure - tel. +44 (0)808 189 2043; email valueflowquality@emergn.com

Emergn Ltd.
20 Harcourt Street
Dublin, D02 H364
Ireland

Emergn Inc.
190 High St, Floor 4
Boston, MA 02110
USA

First published 2012, revised 2021 - printed 16 July 2021 (version 2.0)

Printed and bound in the United Kingdom by Apple Capital Print.

CONTENTS

INTRODUCTION

"And the Oscar goes to..." – this line signifies the climax to a year-round process of planning the biggest event in the film industry. The organisers of The Oscars bear the pressure of delivering a consistently dazzling event year upon year.

Figure 1. A lucky recipient with his award (Helga Esteb / Shutterstock.com)

At the end of each glamorous party, A-listers will polish their awards to take centre stage on the shelf, and hopeful runners up hold out for the following year. But for the organisers, the end of the ceremony marks the first day of planning for the next one. The deadline is exactly 1 year from the end of the event and deviating from this deadline is simply not an option.

The goal of the event is not just to hand out awards to the darlings of Hollywood, but also to deliver a media spectacle to the millions of viewers watching the show on television and as a result – attract advertisers. It's not just an event to praise celebrities; of course, it's designed to make money. Try and put a ballpark figure on the budget for the Oscars... we know it's in the millions, but how many tens would you say? In 2014 it was $40.8 million, so advertising revenue is hugely important to keep up the standard in following years.

Announcement of the nominees marks the beginning of The Oscars season when ballots are sent to over 6,000 Academy members. For the night, photographers and journalists need to have authorised access, the famous statues have to be prepared and of course – the red carpet must be rolled out. 1,500 of the 3,000 guests are VIP's who are provided with an equally important service including catering, limousines, ushers, security, etc. These are just the 'obvious tasks'. Smaller chores include printing invitations, seating plans and preparing the voting ballots.

There is equal buzz around the stars arriving on the red carpet as around the opening of the result envelopes. It requires to-the-minute coordination of 750 limousines followed by interviews among the paparazzi flashbulbs, which basically means a large team hurrying people along at every step. The ceremony lasts around three and a half hours and each speech is limited to 45 seconds so if someone reels off an emotional dialogue, going over their time then that 30 seconds needs to be retrieved later on. Did you see the spontaneous pizza party or the now infamous "Oscar Selfie"? Unfortunately it was not the playful lark we took it for – it was planned in advance. At the Oscars nothing is left to chance.

Event coordinators must constantly assess progress of over 1,000 staff and they rely on feedback to ensure that everything is going to plan – eliminating potential problems so that there are no delays throughout the night. The list really is endless, but every step is essential to deliver the finished end product – a hitch-free awards ceremony.

But, despite the careful planning we must always expect the unexpected. When it's a live event involving so many people, things go wrong. In the last few years not much has happened apart from a name pronounced wrong, or someone tripping on the stairs to collect their award – but there is a backlog of Oscar blunders haunting the internet.

In any project we can plan ahead as carefully as possible, but still have to be flexible when necessary. If a speech runs over at The Oscars, then music is played to encourage the bashful winner down from the stage. This session explores the tools and techniques that currently exist to help us prepare and suggests how we might apply them better.

At the end of this session, you will be able to:

1. Understand the multiple levels of planning in business.
2. Learn to apply planning tools and techniques.
3. Develop existing theories to manage risk and investment through portfolio planning techniques.
4. Formulate a strategy to deal with the issues of planning at scale with high dependencies.

1 THE PLANNING ONION

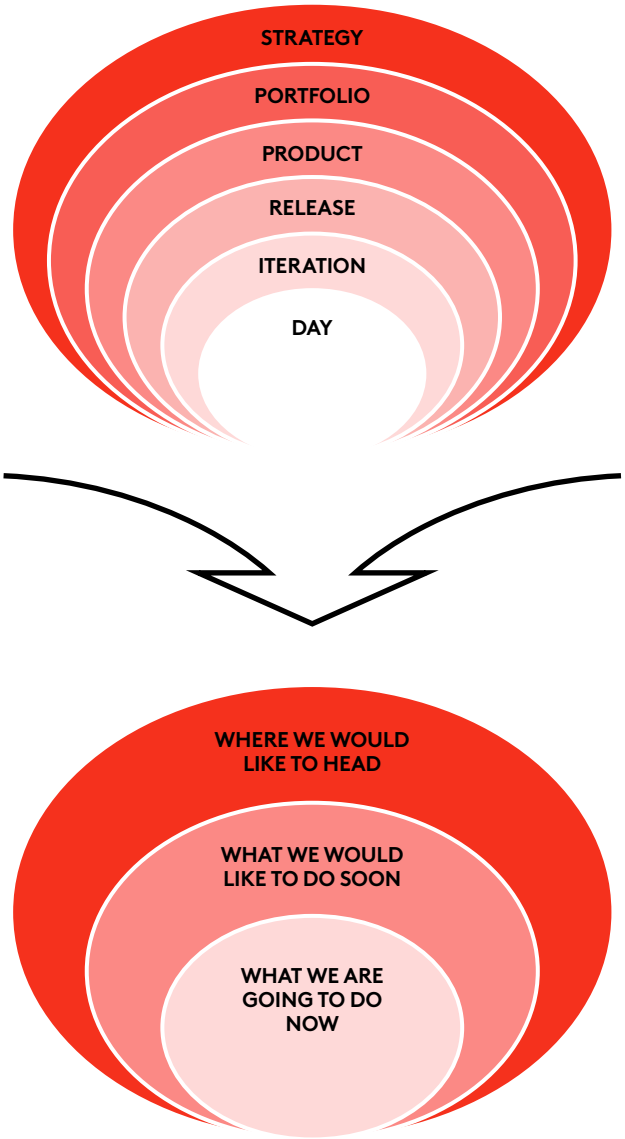


Figure 2. The planning onion and a simplified version

The planning onion is an attempt to formalise multi-level planning for Agile teams. You'll see various different versions of it, depending upon the team, organisation or methodology using it. In reality, the entire concept can be simplified:

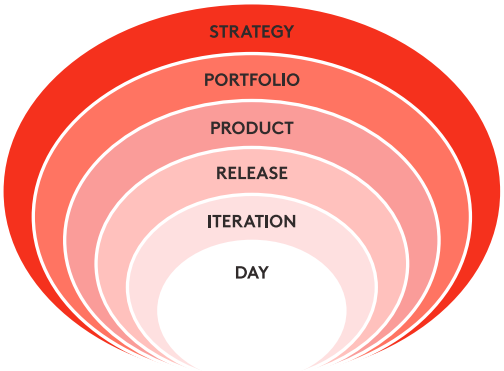
What we are going to do now. This is the area of tactical, detailed planning and the timescale can be anywhere from a day to a couple of weeks.

What we would like to do soon. This is less detailed and directional planning. Depending on how fast you are moving and how far you can look ahead, timescales tend to cover anything from a couple of weeks to the next six months. In some cases you may pre-plan up to a year ahead – for example when it comes to budget for staffing costs.

Where we would like to head. This is strategy and involves setting vision, goals and longer-term direction. There may be no real plan at all, and trying to create any single route to the goal often represents wasted work since the future is subject to such a high degree of uncertainty.

Mike Cohn mentions that 'most Agile teams are concerned only with the three innermost levels of the planning onion.' That is, few teams are given control over the portfolio and strategy planning since this highly strategic level of long-term direction may be under the control of the Board.

1.1. Day planning



Almost everyone begins the day by asking themselves 'what am I going to do today?' This may be formal – a carefully written to-do list, or completely informal – a quick mental run through meetings and phone calls while taking a shower.

Most Agile teams begin each day with a short planning meeting. In Kanban and XP it is known as the stand-up, while Scrum calls it the Daily Scrum. It's role is less as a schedule or progress report and more as a coordination and communication opportunity to highlight problems and allow work to flow smoothly.

The crucial elements of the meeting are:

Time-bounded: Most teams stipulate a 15 minute time-box. This keeps the team focused on what is immediately important – any big discussions about a particular problem should be parked and continued after the meeting ends with those required to fix it. That way other people's time is not wasted. This is also the reason that most teams have the meeting standing up – it makes it clear that you're here for a short time. It also stops people surreptitiously reading email or otherwise distracting themselves from the matter at hand.

Regular cadence: Asking 'who's free now?' or arranging things takes time. These meetings need to be in the same place, at the same time every day. Most teams pick a time such as 9.30am when everyone has settled in, read emails and got coffee but is not yet plunged into a task. Martin Fowler recommends playing Bob Marley's 'Get Up Stand Up' to announce the meeting... we think the choice of reggae is a matter for the retrospective.

Starting the day is not the only possibility however. When there is a wide variation in the teams start times it can make more sense to schedule the meeting just before or after lunch and use it as an energiser.

Everyone present: This central coordination meeting requires everyone to be present. If half the team are off in other meetings then someone will waste time having to catch them up with what was agreed. A well-run stand-up can remove the need for separate status updates or reports and thus save everyone time overall.

Scrum used to have a rule about who could contribute to the Daily Scrum. The development team could speak, while the Product Owner and other stakeholders should only observe (it used to be known as pigs and chickens). The distinction has now been dropped. Essentially as a single team focused on the same goal, anyone who needs to contribute should be there and able to speak.

In a useful location: The point of the meeting is to discuss the work, so the team meets in front of the board or visible workspace. People can move cards around before the stand-up starts in order to reflect what happened the day before. It means that everyone can share the same picture of what's happening.

Structure: Martin Fowler offers the following mnemonic for what a stand-up should provide:

Given Kanban's emphasis on managing the work, many Kanban teams have

---

G - Good start – help begin the day well by providing energy

I - Improvement – support improvement (of both work and process)

F - Focus – reinforce focus on the right outcome (not just busy-ness)

T - Team – reinforce the sense of team and help collaboration

S - Status – communicate what is going on, including new ideas or obstacles

---

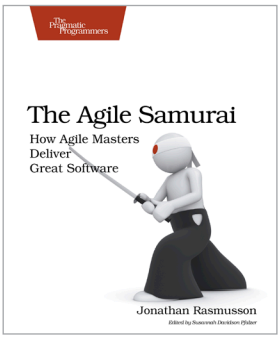
adopted the following method. They consider the cards on a Kanban board, moving them from the input through to 'working' and any other steps needed, such as 'review', then on to deployment. People highlight blockers or problems and agree how they will collaborate to deal with them or whether anything needs to be escalated or requires outside input. Finally, any other business can be raised, including new ideas or bits of information from other teams.

Scrum, on the other hand, has articulated three questions that everyone on the team should answer:

1. What did you do yesterday? (Some teams use alternatives like idonethis.com to capture answers to this and avoid the meeting becoming just a status meeting and instead make it forward facing).
2. What will you do today?
3. Is anything blocking you?

Question 2 is about publicly making a commitment to the team about what will be accomplished – it keeps people focused because everyone knows that on the

following day the answer to Question 1 will confirm whether this was achieved or not. Finally, Question 3 is designed to promote collaboration, in case others need to get involved to help remove the blockers.



Jonathan Ramusson in The Agile Samurai recommended making the wording of the questions more dramatic, including: How are you going to crush the world today? How are you going to blast through any obstacles unfortunate enough to be standing in your way? This, in our humble opinion, may provoke incredulous giggles from most developers, and missiles from a few. With that said, enthusiasm at stand-ups is infectious and may enhance the groups engagement.

In general, we recommend concentrating on items rather than individuals. In other words walking the task board, asking how each item is progressing and what is needed to move it forwards tends to be more productive than asking what each individual team member is doing. This is supported in the saying 'watch the baton not the runner' in Scaling Lean and Agile Development by Craig Larman and Bas Vodde.

There are two reasons. The first is that answering three questions can become rather formulaic and risks the meeting feeling enervating rather than energising. The second is that by asking what someone is doing, it gets easier to confuse busy-ness with progress. What matters is how much value the team is delivering, meaning it makes more sense to focus on the tasks.

Do be careful, however, to ensure that this doesn't mean just two or three people walk the board, leaving others to stand quietly on the edge of the huddle. Whoever is responsible for working on a specific task needs to be the one talking about it – that should mean that everyone speaks during the meeting. Note that the team should be talking to one another – not to a Product Owner or Product Manager.

Activity 1: Improving the daily stand-up

This is an activity to consider with your team – perhaps as part of a retrospective.

Daily stand-ups can sometimes become rather formulaic. The idea is that as well as considering where you are and how you can help one another, you should also use them to boost coordination and the team's energy.

Run a brainstorming session where you consider how you can improve the daily stand-ups for your situation. Steps to consider might be:

Are they being held at the right time?

Do you have enough attendees or are there lots of questions afterwards that have to be referred on?

Are you using them for planning and resolving issues or merely for status updates?

Are you sticking to time – keeping the stand up to 15 minutes or so?

Are you disciplined about keeping detailed conversations for a separate time? Does everyone speak or is the meeting dominated by one or two people?

Is there a better place to hold them – in front of a task board, gathered in a huddle somewhere or just somewhere different?

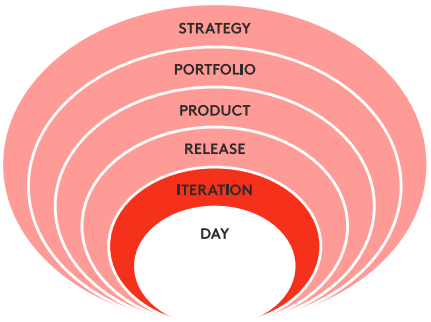
Are you focusing too much on people and individuals as opposed to the work itself?

What do people want to see happen differently? Is there anything that people would find fun and energetic?

Commentary:

The most important element of any such discussion is to ensure that you pick one or two actions, implement them and the review whether they have made a difference!

1.2. Iteration planning



The next layer of planning for the development team looks at the short term future - anywhere between one to four weeks. For most Agile teams this is an iteration, the length of which is matched to the level of uncertainty within the project. Where uncertainty is high, the iteration will be kept short, meaning that teams will be reassessing and planning every week. It provides a form of predictability for users and stakeholders.

The plan simply asks – what is the most important thing for us to achieve in the next iteration and how should we go about trying to achieve it?

Not all Agile methods have formal time-boxed iterations – Kanban, for example, does not. But Kanban recognises the same need to provide predictability in the short term. It provides an input and output cadence and also has a Service Level Agreement (SLA) for differing types of work – these essentially provide a form of time-box. When using Kanban, a regular Queue Replenishment Meeting addresses the same underlying questions: what things will we do next and when will they be delivered according to the agreed SLAs? This essentially functions as an iteration plan.

In Scrum or any Agile methodology which uses time-boxes, the planning normally happens anywhere from a week to a few days before that iteration begins. As well as checking prioritisation, the team must ensure that the requirements selected are ready to be worked on. This might mean estimating value or effort in the light of the most up-to-date information, or it might mean breaking requirements down further and deciding whether extra information might be required. For most Agile teams this will take place in two stages normally, but not necessarily formalised in two meetings. Some smaller teams may do the whole activity in a single meeting – beware though, this can turn into a long, slow activity and thus sap energy.

The first stage is to break work down into smaller items and prioritise them. This may include estimation of both value and effort. The second stage is to agree what the team can complete in the iteration based on the capacity they have available.

Whole team

This doesn't mean the entire team needs to attend any planning sessions, but since iteration planning is a collaborative event, every skillset needs to be represented in order to help define the tasks and provide information on how to fulfil the commitment.

What do we plan to do?

We must ensure that we work on the most valuable items next. This often means updating our prioritisation in light of what we have learned during the previous iteration and feedback from that. For Kanban this will be a discussion between the various business sponsors to select the most valuable items to fill the available slots. It is often important to ensure they strike a balance, feeding in 'intangible' items that apply to longer-term technical issues or non-functional requirements as well as standard or urgent tasks relating to functionality.

In Scrum, this is normally a regular meeting called 'Backlog Refining'. The Product Owner will run through the backlog with several members of the team to select and update the most valuable requirements. It's also an opportunity to get rid of anything that is no longer relevant or useful (this is as important as adding necessary requirements). They will use burndown charts for outstanding work – to determine the work that is left to do versus the time left to do it. It's useful for predicting when work will be complete.

Any gaps in acceptance criteria can be filled in, any missing requirements can be written and the Product Owner can do pre-work in breaking down epics or asking specific stakeholders for feedback or further guidance. Although there will be some elements of decomposition – trying to make sure that the requirements are sufficiently broken down, there is no need to go into any detailed estimation work at this point.

How much can we do?

Running through the selected work, breaking it down, straightening out any knotty problems or dependencies, can take a lot of time. Thus, this is usually done separately and is known as the Iteration Planning Meeting (out of which will come the iteration plan). Scrum calls it the Sprint Planning Meeting. Here, those who will do the work say what they think can be achieved in the following iteration and ensure the work is fully prepared – split down into tasks, estimated, understood and prioritised.

Part of knowing what we can take on depends on knowing what we've done in the past. This requires data. Scrum teams who consider velocity will look at what they achieved in the last iteration or as a moving average of the last three iterations. Kanban teams will be thinking about overall cycle time and looking at the cumulative flow diagram (CFD) for bottlenecks. Whichever measure is used, the idea is simply to predict future performance and speed, based on past performance.

Just because it was sunny yesterday, doesn't mean it won't rain today; completing the crossword in 10 minutes this morning does not mean you will be able to complete tomorrow's, and a velocity of 50 story points in the last iteration does not guarantee the same velocity this iteration. Expectations of what the teams can achieve needs to be kept realistic – neither setting goals too high nor too low.

Everyone likes achieving goals. That means it becomes tempting to set a low goal so that we can over-achieve. It's important to look hard at what we've done, try to stretch occasionally and keep an eye on our realistic capacity.

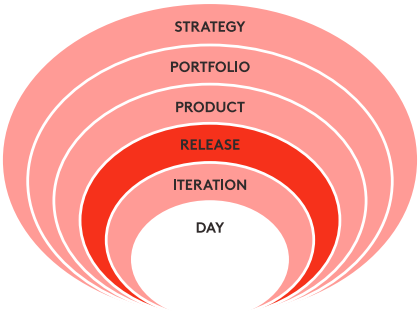
Taking on work obviously requires knowing what capacity the team will have. While in Kanban this may simply be how many available slots are indicated by the WIP limit, for many teams this will mean calculating how many 'developer days' will be available, taking off any holiday, training days or other necessary meetings, etc. This provides an estimate of productive time. Based on the quick look ahead given in the prioritisation meeting, the team should check that if any specialists are going to be required that they will be available.

Formal meetings

Any formal planning meeting should be time-boxed to stop it from turning into a sprawling monster that eats away motivation and unreasonably stops the team from getting on with actual development work. In Scrum, a four-hour time-box is considered standard for a 30-day iteration. Here is a suggested overview of points you might want to consider:

- **Goal:** An agenda helps keep everyone focused. Begin by offering a quick high- level reminder of the vision and where the team are, including new information that came out of the last demonstration or deployment. It is important (see multi-level planning in the Planning session) to ensure that the plan developed for this iteration is going to build against the main plan and keep moving the team in the right direction. Scrum and XP teams like to create a 'goal' that can be displayed and help keep people focused throughout the coming iteration. This can be helpful. 'This month we will complete 40 requirements' is less motivating than 'this month we will be able to process paying customers'.
- **Present the tasks:** Most people should have a rough idea already of what the priorities are. Now the team looks at them and asks if they need to be broken down and who will be working on or owning the item.
- **Estimate:** Never say yes to someone asking 'can you do me a favour?' unless you know whether they're about to ask for a cup of tea or an alibi for murder. Teams need to do some form of estimation to get a rough idea of what they might say yes to in an iteration. It is crucial that this does not become a massive piece of work in which people argue for half an hour over whether a task will take two days or three. See the session book on Estimating and Forecasting for more detail.
- **Review actions:** The Iteration Planning Meeting is the perfect time to ensure that actions identified in the review and retrospective are carried out. Did the team agree to do something different in order to deal with quality issues or amend the definition of done? If so, this is the time to make sure everyone commits to taking key actions in this iteration.
- **Record:** This is a good time to set down any dependencies or assumptions that have been made as part of the planning process – perhaps about external teams or specialists. This forms part of any logistics or communication planning that is required prior to moving into the daily planning cycle described previously.
- **Agree:** The team should now agree what items they intend to deliver (as well as making sure there remains a ready list of items in the backlog that they can pull if they move faster than expected).

1.3. Release planning



The release plan manages to combine some of the really interesting top-line ideas – what shall we release to our customers that they will love? – with some of the most nitty-gritty issues – how do we ensure we don't break anything when we release?

In other VFQ sessions we have discussed how to break a project down into increments. Thinking about what will go into an increment and how or when it will be released is normally called the 'release plan'. A top-line release plan looking at how the whole product might be released in increments will normally be drawn up at inception. This might sometimes be called the 'product plan', just to confuse you.

Figure 3. Release planning at SolutionsIQ

A detailed plan will normally exist only for the first increment or release. The overall release plan or product plan will then be updated as the iterations progress and the team gain feedback on each increment. This in turn, will provide more detail to add to each succeeding release. It is an opportunity to ensure that the development aims for the product are fully aligned with the business aims of what that product must deliver. Where several teams are working on the same product, the release plan will occur at a cross-team level, helping to coordinate their activities and provide system-level decisions on architecture, user experience, etc.

Martin Fowler writes that on an XP project he worked on, 'The release plan is never static and shifts and changes as we learn about what the customer wants and what the developers can do ... but that doesn't stop it from being an essential tool. It gives us a way of seeing what may be achievable, evaluating options, and assessing the effects of events that hit us'.

What will go into an increment means thinking about our customers. Which sections of customers or users will the released product serve? Which of their needs will it meet? Which features are required to meet them? How will the new solution earn money or deliver value? How will it connect to our existing products and services?

All of these questions will end up offering critical choices between what we will include and what we leave out. Tools to help with this include: story mapping, customer journeys, Adzic's burger, and an architectural walking skeleton. You'll find more details in our Delivering Early and Often session. The Product Owner may well need to flesh out the story cards that exist as ideas – holding the necessary conversations with stakeholders and developers to think about ways of meeting the underlying customer need. To this extent, a release plan is as much a creative task as a logistical one.

Functionality desired for a given release should then be matched to important considerations around what is feasible within the constraints of budget and time. Just as with iteration planning, a Product Owner will have to balance features against a predicted capacity.

Remember that much of release planning involves identifying what can be left out. Our aim is to make the release small in order to gather feedback quickly. This means we must be rigorous in thinning features. A release plan will carefully consider what workarounds exist or what tools users can continue using – meaning that only necessary tasks might be supported initially.

Beyond functionality

Features and functionality are not the only consideration. A release plan also thinks about performance questions at a system, as well as a feature level. Reliability, Availability and Serviceability (RAS), scalability, security, user interface, quality and numerous other non-functional requirements will impact heavily on what can be done or not done for a given release.

As well as cutting features out, they can also be cut down, offered at a lower or simpler specification. Rather than just assuming that a feature must exist in a certain way, by looking at it more closely we can identify different performance elements. Jeff Patton suggests asking:

- What minimal characteristics are necessary for this feature?
- What would make this feature more useful in other situations? Should I build in flexibility? How important are the more sophisticated or complex uses where flexibility will be useful?
- What would make this feature safer or more secure? Will there be many novice users? How complex are the rules that must be built in?
- What would make this feature more desirable? Do all the extra features connected to this – make the product easier, more intuitive, more enjoyable – directly contribute to business value?

One technique that Jeff Patton recommends is similar to a story map and with the typical urge for giving ideas different names, Patton refers to this subtly different technique as a 'span plan'. It is a way of visualising which tasks are necessary to support all the business activities in a process. This forms the basis of a release plan.

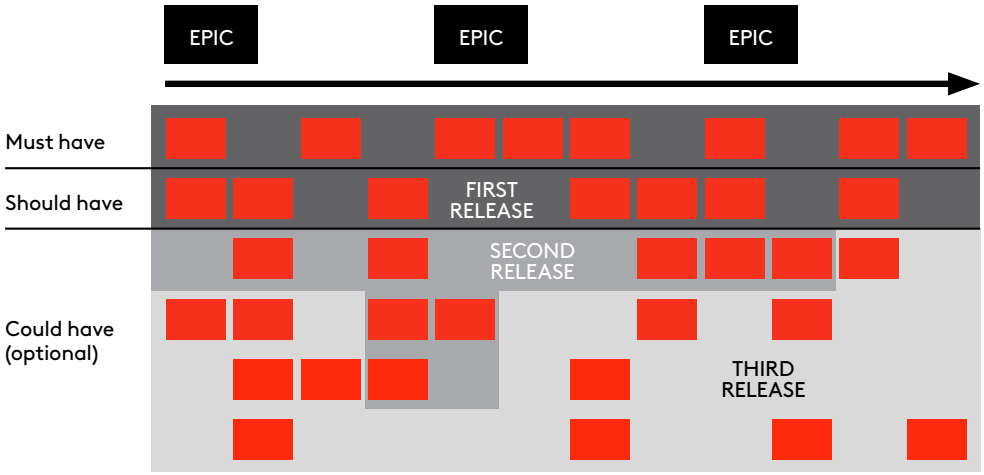


Figure 4. A span plan showing stories to be included in each release

Activity 2: Build your own span plan

This activity should be completed either alone or with your team.

Start by identifying the key activities in a process. We've selected Order Processing as an example. Obviously, you should use your own project.

- **Activity 1** – receive order from shop
- **Activity 2** – check for stock
- **Activity 3** – identify customer and retrieve account details or register and credit check new customer
- **Activity 4** – generate invoice and apply discount
- **Activity 5** – send invoice, order and delivery instructions to warehouse for dispatch processing

On a board draw a vertical axis with Necessity at the top and Nice-to-have at the bottom. The horizontal axis will show the user's linear progression through the system.

Now begin identifying all the tasks that make up each activity and run through Jeff Patton's questions to identify the necessary tasks that support the business activities. Remember to think about workarounds and alternatives. Begin to place your tasks in order on the board, below the relevant activity and prioritised by necessity.

Now take horizontal slices through the system. The first release must include all tasks necessary to support the business activities. Subsequent releases can add greater support.

Releasing with the release plan

Release planning needs to take account of whether your code is truly ready for release. Now of course, we advise continually building and integrating your code, and having predefined release scripts. With your strictly defined understanding of done, all your code should be fully tested with all necessary user documentation or manuals, while your team have mocked up every potential internal and external connection. With that, at the press of a button you should be able to release your product.

Hmm well... Life isn't always perfect. Perhaps there is a historical legacy of deployment issues, perhaps the necessary work to manage continuous integration was simply too expensive for this project. Whatever the reason, for many companies there is little confidence that the code can just be released. They need to build in a period of 'hardening' or 'stabilisation' as it is also sometimes known.

What's wrong with hardening?

Agile can sometimes feel a little judgemental on this subject. It's all very well to know that you ought to be continually integrating, but if you haven't, then simply being told off for this is not helpful. The real problem with hardening is that it builds up risk. Deferring technical debt doesn't make it magically go away – and it means you are blind to one of the biggest sources of uncertainty there is. You don't know what problems will arise in this phase and therefore you have no idea how long it will take. Your release date could suddenly slip by three months or more – with no prior warning!

When release planning you must think ahead and work out what elements of a hardening process can be done earlier, and what cannot be done until the end. These issues need to be addressed from the first release plan onwards. A general rule of thumb is to try to bring forward the things that could stop us releasing. This might not be possible on day one, but something that teams can aspire to over time.

Risks exist with Agile

Even if you've done everything right, it doesn't mean that everything will go right. Release is the moment that your product leaves the artificial world and enters the real world. There are plenty of things that can go wrong. There are plenty of things that you don't control – from users themselves to external systems. A mock-up of how your system interacts with another system is not the same as the real thing. A mirrored production domain is still not the real thing. The potential for something happening that damages the organisation (down time, impact on customer data or service, etc.) is high. Even if you have automated every step of deployment, you can't automate people or their behaviour. Much of your release planning process will thus be devoted to talking to people and finding out what might happen and then deciding what plans you will have in place to deal with those particular scenarios.

Some issues make small and frequent releases difficult. We don't underestimate the problems these cause. Consider a system change that will affect hundreds, perhaps thousands of users. When will they receive training on the changes? If the change will affect how they do their work then will the changes be mandatory or can they accept them when they're ready?

The place to discuss the risks, benefits and potential disruption of such problems is the release plan.

And another thing...

Below we have looked at the kinds of activities and risks you need to consider in the release plan. These occasionally get forgotten and are pushed into the dreaded 'stabilisation phase'. It is best to pull all of these (and of the many activities you already have that are unique to your particular context) into a checklist for the team to go through in the Release Planning Meeting and keep as part of the plan.

Any user documentation: Installation guides, user manuals, deployment guides for third parties, etc. Much of the preparatory work should be done as the project goes along, but the final layout, proofing and issuing may legitimately take place at the very end.

Checking the product/release with other internal people: A lot of this is talking and coordinating. Don't underestimate how long such activities take! You may need to liaise with other departments, including marketing, sales, deployment, operations, customer service, etc. and find out what they are worried about and how you will take their concerns into account. There may also be formal rules around governance and internal standards. If, for example, there must be a period prior to the release window where no changes can be made, then the team will need to branch the code. The team will need to create a process around this for when it happens and how it is stored, etc.

Checking the product/release with external parties: Does your change interact with another system or component? This could be anything from a client server to a service like PayPal. The interaction and integration must be tested, and you will need to check you comply with any governance about rolling changes back or outages, etc.

Testing: Yes, we know you've been testing as you go, Agile crew, but there is always more testing before launching a flagship product or mission critical process change! As well as all the system integration, test scripts and regression testing, make sure that your release plan allocates time for exploratory and field testing, as well as system-level tests and quality requirements testing for all the non-functional requirements (remember to think about potential scale problems).

Localisation: The need to support localisation issues, from language translation to re-sizing forms, etc. can sometimes get lost in the hard focus on producing functionality. The release plan keeps an eye on when and how these types of work will be done throughout the increment plan, rather than leaving it all to a panic at the end!

Innovation: This might seem a strange heading to include in the checklist, but actually the release plan is a good place to ensure balance – not just of necessary features, functionality and performance attributes, but of innovation. Agile teams pride themselves on achieving a 'sustainable pace', but when the pace is geared towards churning through iterations to provide the smallest possible 'necessary' increment, sometimes innovation can get left behind. Release plans often include some time for research spikes or days to just try things out and see where the ideas might go.

Feeling the pain

There's a rule in product development (the true home of masochism) that if it hurts, you need to do it more often. If the decisions and risks of frequent deployment are slowing you down then the team needs to try and work out how to make it easier.

This might mean automating elements of the deployment process or investing in the infrastructure to do seamless upgrades that users don't notice. It might mean simply refining the checklist and the process until a weekly push is something that everyone just takes in their stride. Only you know what are the pain points for your specific context and thus only you can create a plan to improve on these.

Just remember that when Toyota instituted their Production System, it was a matter of necessity. Changing the line over from one type of car to another took a full 24 hours. They gradually reduced the time required. By 1962 the company average was 15 minutes. By 1971 they had whittled this down to 3 minutes. None of this was a case of just 'getting faster'. It took creativity and intelligence to reduce the set up time. To select just one tiny element: the dye-cutters that cut the metal used to be sharpened by the machine operator. This work was then taken 'off-line', meaning the factory had two sets of cutters. While one cutter was operating in the machine, the other was being sharpened by a 'grinding' department. At the changeover, the new sharp tools would be slotted in and the old blunt ones taken away. By investing in two sets of tools and the extra grinding specialist, the factory gained almost 20 minutes of changeover time.

We need to use the same creative thinking to pick away at the problems that slow down our development by running tests in parallel, for example, or investing in the most appropriate hardware. Whatever the challenges your team face, forcing yourselves to bite the release bullet – like Toyota – offers considerable long-term gains.

Activity 3: The planning game

This technique was first developed in XP as a way to 'create a little emotional distance' for the occasional clashes that occur when differing parties come together in planning. It can be used at the iteration level, but is probably most useful in release planning.

To those whose only idea of gaming involves a gun and a fast car, this is going to be a slight disappointment. Bear with it. This is a cooperative game with the goal of putting the greatest amount of value into production over the life of the game.

Players:

- **Stakeholders** – they have the most information about what is valuable and important, so they prioritise.
- **Development** – they have the most information about what it will take in terms of time, effort or money to deliver something, so they estimate.

How to play:

1. Anyone creates a requirement or selects a pre-written but unplanned requirement.
2. Developers estimate the requirement. If it can't be estimated, then the requirement must be split until estimating it is possible.
3. Stakeholders/customers place the requirement on a board according to its relative priority.
4. Continue steps 1-3 until all requirements are placed and no-one can think of any more to write.

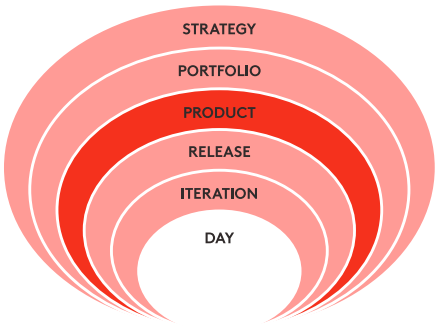
At the end you will have a single list of cards in priority order. It is impossible to 'tie' two requirements of equal priority; one must be placed before the other, even if the ordering is random.

Commentary:

Anyone can ask questions about an estimate or a priority order, but each group has the final say over its area of expertise. The process of questioning is pretty important! When a stakeholder asks 'why is that so expensive?' some crucial design options might emerge as people dig into what's difficult about creating the existing feature.

The game can also raise disagreements between stakeholders over which set of features is the most important. This might feel uncomfortable but it is better to explore this now, than to allow two people to think they're going to get what they want – only for both to be disappointed.

1.4. Product planning



In product planning there are several tools that make it easier to lay out and think through a plan to introduce a product to your users and to keep responding to their needs over time.

Product backlog

A well-structured product backlog is at the heart of any well-functioning team. It is a list prioritised to ensure that the team works on the most valued features first. It should contain descriptions of all of the functions needed to complete a project. Typically, a team will start by sitting down together with the Product Owner and writing down all of the things they can think of for the backlog prioritisation. It commonly includes requirements, bugs, research tasks and engineering improvements. Each item is estimated in units that are often called story points. This will always be enough to get a project started.

The team estimates when each work item will be complete and relays this to the Product Owner to give clarity on timeframes. The backlog should then be treated as a living dynamic prioritised list, allowed to grow and change as more is learned about the product and the customer. All work carried out by the team should come from the product backlog; it is the front door to a project. If it includes all work then the Product Owner, the team and management have a better picture of the work that remains and it reduces the chance of any last-minute hitches.

Managing the product backlog is no easy task; in fact it's a full-time job. There are a few simple techniques that can help with this – changing an overwhelming and time consuming task in to an interactive and iterative process.

A well-developed product backlog does not ensure the desired outcome, but if it's lacking then this could result in an incomplete product that will not meet the requirements of your customers and stakeholders.

The Business Model Canvas

One of the popular tools for managing a strategic plan is the Business Model Canvas, which offers a template of nine basic business building blocks to encourage an organisation to think about their business model. A slightly different version is known as the 'Lean Canvas', which keeps things simple and brief, as it has been adapted to the ideas of Lean Startup. This really bridges the gap between strategy level planning and product planning as the canvas covers our plan across all levels.

Whatever tool is being used, in general the aim is both to sketch out where the company is and where it intends to be. The difference then forms the basis of the plan – which activities the organisation should concentrate on in order to get to where they want to be. The team should also note any assumptions that this direction is based on – whether numerical or otherwise.

The canvas should preferably be drawn up on a large sheet of paper or white board so that several people can contribute together – placing Post-its or scribbling ideas down. As always the key is to keep the canvas dynamic – changing it as you gain feedback, testing the assumptions and updating or adding to figures as you learn more about the market and customer-base.

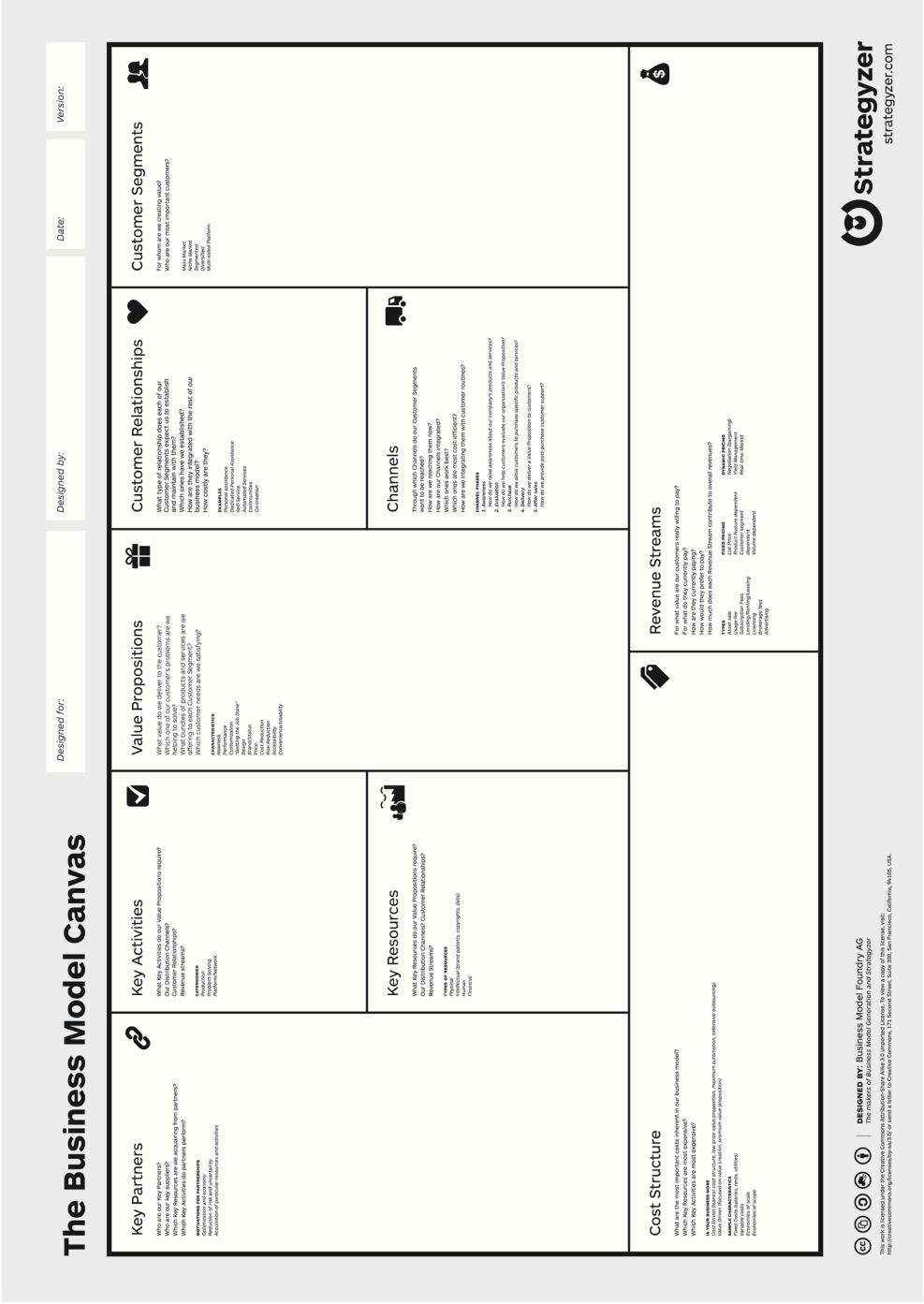


Figure 5. The Business Model Canvas template

Customer segments

If you have a multi-sided market then you will have as many markets as sides. So for example, an online media company will have readers on one side and advertisers on the other. The segments are the 'macro' analysis and your individual customer base is the 'micro'. Visualise each micro as a customer persona – you should understand what they think, see and feel about your product or service. Consider the user and the buyer of your product or service (if they are different). If you have more than one customer persona then prioritise them in a list. Write down your job to be done for the customer – what's the need that you're fulfilling? Pairing customer segments and value propositions is the driving force for everything else in your model.

Value proposition

From your list of personas note the problem or need that you are fulfilling for the customer, and what is unique about your value proposition compared to the alternatives out there. Make a prioritised list of value propositions that link to each persona.

Channels

We communicate our proposition to our segments through channels and also sell products or service customers through these. Note what steps are needed for each of sales, promotion, service, etc. Consider whether your channels will give enough visibility into the user.

Customer relationships

Consider the interaction with your customer through a sales and product lifecycle. Can your value proposition be delivered to the customer effectively through a channel? Does delivering a service over the phone work or is it worth charging a premium for personal support? If you are exploring a new venture then be sure to document and review critical assumptions here to test your customer relationships model.

Revenue streams

Map segments to propositions, to revenue streams so that you can see the links on your canvas. If this is a new venture then look at where you are driving revenue and whether it is aligned with the focal points of your plan. Are you charging on value or perceived value?

Key activities

The crucial things that your business needs to deliver on its propositions and make the rest of the business tick should be outlined in this section. If you are a product-focused business then this will include ways of building a better product and continuous learning about your user. If it's a service for a particular set of customers then it may well include maintaining expertise on a segment or creating products or services for that customer.

Key resources

These are the strategic assets that you need in place – and to a greater degree than your competitors. There are three core business types proposed in the Business Model Canvas – product, scope and infrastructure. In a product-driven business there will typically be key talent in critical areas of expertise and property related to their offering. In a scope-driven business the typical need is for key knowledge on their segment, a repeatable set of processes and sometimes infrastructure. For an infrastructure-driven business they achieve economies of scale in a specific, highly repeatable area so types of physical or virtual infrastructure are the key resources.

Key partners

It is useful to map your key partners to your key activities to denote which specific partners are handling which key activities for you. Note down a description of the relationship between these.

Cost structure

You will already have worked out how your key activities drive your propositions and therefore your revenue. Now work out how they drive your costs. Check that these are well aligned to the key value propositions.

The canvas does a great job to help figure out your business plan. Of course this is just the starting point to a project or business venture, and we naturally go on to look at maintaining a long-term competitive advantage. There are other models we can use alongside the Business Model Canvas to move in the right direction, such as Porter's Five Forces Analysis. Go through the steps and see how this hangs together with your Business Model Canvas. Another simple Lean Startup tool is the Value Proposition Canvas, that gets you to the 'minimum viable clarity' required to start building and testing.

Activity 4: Using the Lean Canvas for your product

Longer-term planning tools can be used at a product, category and company level. It can be a good exercise at the beginning of a new product, for example, to think about the core customer and usage in order to decide on the MVP and key assumptions. Or at a product's launch, this might help you think about which features, upgrades or variations should be planned next. It can be applied to any part of a company's strategy such as an update or change project, as well as actual products.

Take the Lean Canvas template on the next page and begin to fill it in for your product. Ask a couple of team members to help you along with the Product Owner and see how it deepens your understanding of the product and its importance to the organisation.

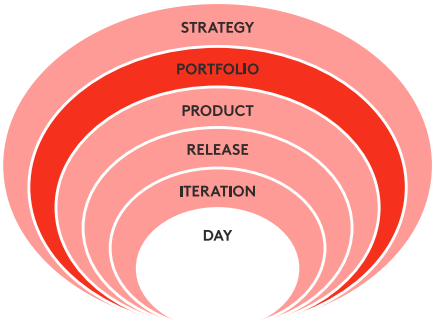
In the Lean Canvas template example below, we have filled it in for a child's lemonade stand to give you an idea of what it might look like.

<table>
<tr>
<td rowspan="2">PROBLEM:
What is the central problem for a customer that you would like to solve?
People are thirsty on a hot day</td>
<td>SOLUTION:
What product or idea will solve this problem?
Tasty, homemade, iced lemonade</td>
<td rowspan="2">UNIQUE VALUE PROPOSITION:
Why would someone want your solution?
It's made fresh from real lemons within the last hour and it's perfectly chilled</td>
<td rowspan="2">UNFAIR ADVANTAGE:
What's the secret ingredient that should convince someone to buy or will stop a competitor copying it?
Granny's secret lemonade recipe passed down through the generations. Ideal location on the side of the street outside our house – where there are no cafés of shops</td>
<td rowspan="2">CUSTOMER SEGMENTS:
Who are the main groups of people likely to be your customer - are there different targets or niches?
Any one walking or driving along our street who is thirsty. Key time will be the return from the school run at 3.30pm</td>
</tr>
<tr>
<td>KEY METRICS:
What will you measure to see if you are on the right lines?
How many people buy the lemonade, how many return for a second glass, what people say as they drink the lemonade</td>
</tr>
<tr>
<td colspan="2">COST STRUCTURE:
What are the major costs in bringing you idea to market?
Fresh lemons, sugar, ice, cool box, cups</td>
<td colspan="2">CHANNELS:
Where will people find/ buy your product?
On the pavement outside our house</td>
<td>REVENUE STREAMS:
How will money come in for your product?
We will vary our prices depending on the weather. 50p normally, 70p on a hot summer's day. We will charge 10p less for a refill</td>
</tr>
</table>

PROBLEM:	SOLUTION:	UNIQUE VALUE PROPOSITION:	UNFAIR ADVANTAGE:	CUSTOMER SEGMENTS:
What is the central problem for a customer that you would like to solve?	What product or idea will solve this problem?	Why would someone want your solution?	What's the secret ingredient that should convince someone to buy or will stop a competitor copying it?	Who are the main groups of people likely to be your customer - are there different targets or niches?
	KEY METRICS: What will you measure to see if you are on the right lines?		CHANNELS: Where will people find/buy your product?	

COST STRUCTURE:	REVENUE STREAMS:
What are the major costs in bringing you idea to market?	How will money come in for your product?

1.5. Portfolio planning



The first three layers of the planning onion are clearly about horizons of time – a day, an iteration, a release. When we get to the product planning layer, though, the focus shifts from time to things: products, portfolios, and strategies. Strangely these are still time horizons but they are not as easily marked by start and finish dates, and they emphasise other aspects of planning. A portfolio plan can be used in many ways; it can apply to a portfolio of products, a portfolio of features or even tasks. The true purpose of creating such plans – and the nature of portfolio planning – is to manage investment and risk.

There are usually more good ideas than we have resource to implement them – and this is true at a feature level and also at the macro, product or project, level. In fact the list of things that we might be able to improve, automate or innovate on is almost limitless in its possibilities. Any organisation needs to think about how it will balance the investment in business initiatives (whether IT or not) in its portfolio.

Portfolio management selects which ideas will be funded and to what extent. This will normally require an element of balancing out high-risk innovation against low-risk maintenance or perhaps expensive but essential upgrades. Within a product it might be about balancing the development of exciting features (but which might not work) with the bread-and-butter account work required or offering parity with a competitor. We may refer to this as slicing through the increment and value prioritisation, but in essence the techniques are the same.

Portfolio planning is a good method for investors and shareholders who need to balance their risk. You might do it yourself when considering your pension – perhaps when you're young you'll have some high risk and high growth elements as well as more stable securities, but as you approach retirement, you'll probably want to move your portfolio into a more conservative, low risk series of investments. You'll probably spread your risk across geographic regions as well, investing in Asian, European and American markets.

The problem is that such an approach doesn't always apply to companies. There are many times when 'balance' and minimal investments will not work for a company; times when a company has to embrace risk, invest heavily or concentrate in a single area to achieve market dominance. It can be more relevant to think about a company balancing differing levels of innovation, of revenue protection with revenue growth initiatives, of increasing capabilities with reducing costs.

It's important that this happens at a level beyond departmental areas – you don't need a turf war between IT and marketing over who gets a bigger budget to deliver innovation. The business needs to continually evaluate how an idea will impact on the organisation as a whole. That means that all projects must be assessed for their fit to the organisation's goals and strategy; their risk profile, their projected value return and how difficult or expensive they may be.

Having provided and quantified the results, projects can be ranked in order within the portfolio (read the Prioritisation session for more detail). This ranking needs to be reviewed frequently – managing a backlog for investment decisions, just as a Product Owner would manage the backlog in an iteration.

Naturally, in doing this we need to avoid falling into the trap of doing enormous amounts of up-front design. This is fairly difficult, right? Given that you need to answer the questions: how much might this cost me and how much might it be worth? In order to come up with numbers that are anything better than a guess, then work needs to be done in advance – leading us straight into a planning paradox.

Simon Baker from the software development lab Energized Work, looks at the question the other way round: what are you prepared to lose if you turn out to be wrong about the business value? It follows on naturally from what we cover in the Estimating and Forecasting session – the process of making a small investment in order to learn more or test demand. It requires a rolling programme of investment decisions, whether we will stop investing, invest more or invest and change direction.

Game theory and strategic decision making

The most famous decision-making experiment in game theory is called the prisoner's dilemma. First formulated in 1950, the dilemma is that of two prisoners, both accused of a crime and held separately. Each is offered a bargain: confess, and you will face two years in prison; stay silent and you will serve just one year. But – if one of you confesses, and the other stays silent, then the one who confesses will go free, while the other will go to prison for three years.

The logical outcome is that each, self-interested prisoner should confess, hoping to betray the other. And yet, this outcome, with both betraying, will be worse than if they had managed to cooperate and both stay silent.

What has this to do with decisions in our planning? The different possibilities equate to ways of looking at our risk and the choices we make – our strategy.

The prisoners are trying to minimise the potential downside (time in jail) and they choose what is called a 'minimax' strategy. Organisations occasionally need to take minimax strategies where they avoid possibly destructive risks. They need to minimise their exposure to disaster, but can also maximise their exposure to the unexpected positive benefits. This is the minimax strategy that we've described as part of portfolio planning – let's make lots of little bets and hope that one takes off while minimising our investment in the many ideas that fail.

The maximax strategy is where we invest everything in search of the big pay-off. That may sound wrong, but in fact it is the correct approach in some circumstances. Steve Jobs referred to it as a 'betting the farm' moment. Sometimes this 'maximax' return, investing everything with maximum risk for maximum return is the right, and perhaps the only, thing to do. This is the area of crises, of brave decisions and also of disasters. There are times when an incremental approach is simply insufficient. The case study below is an important reminder of why we should not allow Probe-Sense-Respond to become a new orthodoxy.

CASE STUDY: US forest fire management

As part of its forestry management, the US Forestry Service tries to control fires to protect trees. This used to mean a policy of stopping all fires – removing all risk. Unfortunately, it turned out to be misguided.

Wildfires burn naturally in many forests. Trees like the Californian sequoia have evolved to deal with them. This tree has a fibrous bark up to 10 inches thick which is highly resistant to fire. Meanwhile the lowest branches are several metres high – out of the reach of most wildfires. In fact, the periodic fires help clean out the brush and litter along the ground, as well as any smaller specimens (meaning the larger trees face less competition).

When all fires were put out – as was forestry policy at one time – the dead wood and brush built up along the forest floor. As we know, there is no such thing as certainty, meaning that eventually, in spite of all precautions, a wildfire did happen. Now, with all the extra fuel – the fires burned higher and hotter than was normal; so high that the branches of the sequoias caught fire and whole trees went up like torches. After this, the fire could spread from treetop to treetop, carried by the breeze and through branches less well protected by thick bark than the sequoias' trunks.

Figure 6. Smokey Bear warning of the risk of wildfires

The Forestry Rangers now set small 'prescribed' fires in an attempt to offer the normal protection of natural wildfires and manage the forest better. There is still, naturally, the potential for uncertainty. In the case of the Los Alamos Fire in 2000, a prescribed fire got out of control and burned 48,000 acres and destroyed $1 billion of property.

As soon as the fire began to spread, the firefighters reacted decisively. A 'minimax' response of adding one firefighter at a time and seeing what happened, would have been disastrous! Instead the team went to a 'maximax' response. They mobilised over 1,000 firefighters and made the decision to evacuate over 18,000 people from potentially affected areas.

Programme Portfolio Management

In an organisation there is always a collective of people who will share the highest level of decision-making responsibility. SAFe name this group of people Program Portfolio Management (PPM). This represents the business managers responsible for strategy and investment funding, programme management, and governance. Compared to the rest of the organisation, these people have the best understanding of the overall business strategy, technology and financial constraints and defining and implementing the portfolio product/solution strategy. Often they are assisted by a Programme Management Office (PMO) who help to guide execution and governance. By managing the key responsibilities this team establish and communicate objectives that guide the company's investments and strategy. Alongside, they create a vision for the portfolio and help determine value streams, allocate budgets to teams, define and prioritise the portfolio backlogs, and report to the business on investment spend and programme progress.

For our organisations to be successful we rely heavily on a strong PPM strategy. However, over the years certain methods have caused us to adopt behaviours and mindsets that prevent us from effectively adopting Lean and Agile ways of working. To change our mind-sets there are seven transformational patterns to move your organisation to a Lean-Agile way of Programme Portfolio Management.

From traditional approach	To Lean-Agile approach
Centralised control	Decentralised decision making
Project overload	Demand management; continuous value flow
Detailed project plans	Lightweight, epic-only business cases
Centralised annual planning	Decentralised, rolling-wave planning
Work breakdown structure	Agile estimating and planning
Project-based funding and control	Lean-Agile budgeting and self-managing Agile release trains
Waterfall milestones	Objective, fact-based measures and milestones

Figure 7. Seven transformational patterns for Lean-Agile Program Portfolio Management (from scaledagileframework.com)

Using these patterns we can better understand how to fulfil our primary responsibilities, facilitate and implement our business strategy and investment funding, and enhance programme management and governance.

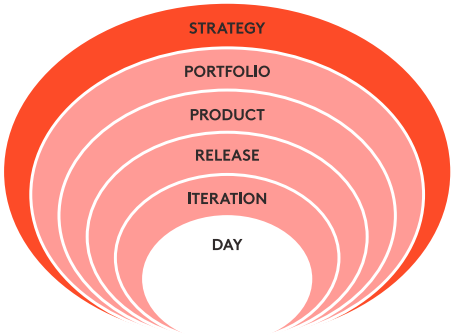
Activity 6: Examine the portfolio

This activity requires you to work alone, asking colleagues for information as necessary.

Consider the projects your company has worked on over the last year. Can you identify which strategy each one followed? Was it a minimax or a maximax strategy? From the projects aims and risks can you determine if you think the strategy was correct? Be sure to examine whether behaviour matched rhetoric! Often companies claim that they are investigating value, when really their behaviour suggests that they are simply increasing the size of their bet!

Informing the various Product Owners or senior management that they are doing it all wrong is unlikely to win you many friends. Instead, we suggest using the data you have gathered to help in deciding the approach to any new projects which you are working on and to back up an argument as to why you wish to test an assumption or scale up quickly in the face of a severe risk.

1.6. Vision and strategic planning



Anyone can set a goal: 'Kill Microsoft'; 'Provide the most effective computing solution'; 'be loved by our customers'; 'Crush Adidas'... Whether it's the right goal is a different matter. But this is not a plan. In fact, without a strategic plan on how to reach it, a goal is just a wish.

The idea is that through a strategic plan we can check that all the activities we do are leading us in the right direction. If we want to be the world's leader in premium gold jewellery but we keep importing plastic watches because they sell well at Christmas then we either need to change our strategy or change our activities. There's absolutely nothing wrong with selling cheap, poor quality products – although you might dress the strategy up as 'offer customers fun fashion at fabulous discounts'. But if our strategy and tactics are completely misaligned then we risk diverting our energy and confusing our customers with mixed-up offers, communications and products.

There are numerous examples of companies who have ploughed millions of dollars into innovations at odds with their brand and their strategy. Pepsi's Crystal launch is less frequently quoted as a product disaster than Coke's infamous 'new' flavour, but it proved a costly embarrassment to the company. In concentrating on 'natural' and 'new age' imagery and language to produce a clear version, Pepsi had fundamentally misunderstood their customers and their main brand. People who drank Pepsi wanted the original drink; those who were into more 'natural drinks' didn't want a carbonated sugared drink whether it was clear or brown in colour. David C. Novak is credited with introducing the Crystal Pepsi concept. In a Fast Company interview, he stated: 'I still think it's the best idea I ever had, and the worst executed. A lot of times as a leader you think, "They don't get it; they don't see my vision." People were saying we should stop and address some issues along the way, and they were right.'

Figure 8. Pepsi Crystal - clear cola

Pepsi is not alone. The Bank of America, whose corporate goal is 'to improve the financial lives of our customers and clients', announced they would begin charging customers $5 a month to use a debit card or withdraw cash. The reaction was enormous, provoking an unprecedented Twitter storm and direct criticism from the President. The bank eventually climbed down, but not before it had lost significant numbers of customers and endured a PR disaster.

Figure 9. The caption they ran after reversing the decision

The danger, and it's a danger seen in many corporations, is that creating the strategic plan can become a very internally focused activity, involving a lot of time and money expressed in numerous PowerPoint presentations and beautifully crafted statements but with little connection to reality. The best strategic plans remain simple (like the goal) and help everyone in the organisation make decisions.

Google were successful in implementing a plan to achieve this through a system called Objectives and Key Results (OKRs). It's simply a framework to get things done – by setting an objective and outlining a number of key results to help reach the objective. Google set up OKRs at a company level, a team level, a managerial level and a personal level so that they can all work towards a shared goal. The results are graded at the end of each quarter to determine how close each person gets to their individual goal. They then use the results to decide where they need to step up, what they are doing well, or indeed, whether the key results they set themselves were too easy. The OKRs are on a database, which is accessible to the whole company. It may sound intimidating, but Google have found that this system is a quick and easy way to view what each person is working on, and to help them do their job to the best of their ability.

So this is how Google work; we need to find a way that works for us. When a new opportunity comes up, people on the ground should be able to measure the idea against the strategic plan and ask key questions: Will this idea help us achieve our goal? Will it clash with or divert attention from more important work?

Asking and answering these questions honestly can help organisations find the courage and clear-sightedness to turn down tempting short-term opportunities and remain focused on the mid- to long-term. Of course, there are times (like with a small start-up or during crisis) when an organisation has to take a very short-term focus in order to survive, but in general, lurching from crisis to crisis is a poor way to build a successful business.

Long-term planning tools

As we said in the Planning session, situations can change at a minute's notice and catch us when we have no plan in place. Scenario planning is used to make flexible long-term plans. It's really a way of exploring the future – if you are aware of what could happen then you'll be better prepared for it.

Scenario planning exercises involve identifying trends and exploring the implications of projecting them forward. These are usually recorded as high, medium and low forecasts of political, economic, social and technological (PEST) trends. You'll see that as different trends are chosen and combined with the different levels of forecasting, you will identify a whole range of possibilities.

Although it doesn't guarantee your ability to deal with a situation – carrying out regular scenario planning exercises makes you more aware of the possibility. It therefore enables you to act rapidly if a situation does develop.

Resisting change is part of human nature – the mind struggles to recognise information that it doesn't understand or expect. So scenario planning will also sensitise people to information that they may not have noticed otherwise.

As a management tool, scenario planning exercises involve letting go of preconceived ideas and expectations. They require us to be creative in group 'dreaming' sessions. Many soft tools are useful such as brainstorming, PEST analysis, systems analysis, study of past technological changes, analysis of historical discontinuities and moments of change as well as the 'unintended consequence'.

CASE STUDY: Making strategic plans real

No company in the world comes right out and admits that they offer 'awful customer service, but cheaply, so don't complain'. Although some might argue that Ryanair comes close. The temptation is to have lots of fine sounding 'strategic goals', and then – like the Bank of America and many others – ignore these completely in the actual plans that are made. Some companies try to ensure they don't do this by building in mechanisms that keep the goal front of mind.

Graniterock – a small company selling gravel, sand, concrete and asphalt – set themselves a goal of providing 'total customer satisfaction'. That sounds like a phrase that could be read in many strategy papers – so the co-presidents decided to make it real. They instituted a 'short-pay' policy. Every single invoice from Graniterock had a line at the bottom that offered the customer a chance not to pay all of the invoice. It read:

> "If you are not satisfied for any reason, don't pay us for it. Simply scratch out the line item, write a brief note about the problem, and return a copy of this invoice along with your check for the balance."

This was not a marketing gimmick. Many firms have used 'satisfaction guarantees' where if the customer is not happy they can send back the product and receive a refund. Such offers tend to rely on customers not wanting the hassle of organising the refund. Here the customer could keep the product and simply not pay – with no questions asked. It was a risky strategy – but it certainly focused everyone in the company on customer satisfaction. Feedback could not be ignored – because poor quality or poor customer service meant no money. Uncovering and fixing problems became a top priority. The policy is a big part of Graniterock's reputation for quality and customer service – and the premium they are able to maintain in a commodity industry.

1.7. Limitations of the planning onion

So we know where we're headed. We know what we're releasing next. We know what we're doing for the next two weeks and we know what we're doing today. What could possibly go wrong?

The problem is that frequent planning and a short planning horizon still doesn't provide us with the answers to some very important questions; questions which are perfectly fair to ask. Things like – how many staff do I need to hire? How do I know whether to start development when I don't know how much it might cost? When can I start planning a launch?

For all our insistence that it's not possible to answer with any certainty, you still need to make some assumptions based on those questions. For example, if the product will take six people to build and we only have three then we need to start the recruitment process now.

CASE STUDY: 22cans and the curse of popularity

The game design studio 22cans launched a free app with a game called Curiosity.

Billed as a 'social game' with a prize that would be life-changing, up to a million users attempted to access the server. The results were appalling – the app became slow or failed to work at all – which, given the interactive, multi-player nature of the game first frustrated and then infuriated players.

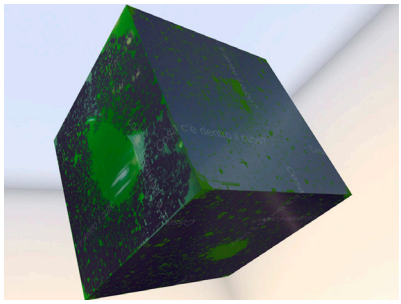


Figure 10. A screenshot from the game

Although 22cans developers worked all night and in spite of paying massively to increase server capacity from 1 to 14, bugs and performance problems dogged the team. Eventually, the company had to ask for donations to help with the server costs (as well as issuing several apologies). It was exactly the kind of problem that might be expected to happen to a lean, small start-up, but they didn't receive too much sympathy from players and commentators. 'Under-delivered and over-promised' were some of the more repeatable comments on game sites.

But what should they have done differently? No-one would have been impressed with 22cans if they had bought 14 servers (on borrowed money) to support a game that was a flop. They could perhaps have tried for a soft launch with less PR, but they were probably nervous about having too few gamers rather than too many. They could also have created contingency plans to deal with the 'risk' of massive success and immediate heavy demand, but at the time such plans probably felt wasteful and presumptuous to a team focused on getting the MVP launched.

2 PROJECT PLANS

If you're like most organisations, you create and work on projects. Projects have plans! As we have learned from the planning onion it may make sense for a project plan to cover a single iteration or, perhaps, a single release. In either case we now have a few more things to consider to form a planning perspective: sequencing of activities within our time horizon and managing dependencies we have on other projects.

Dependencies and the critical path

Almost anyone who has ever worked in any kind of product development is familiar with the concept of the 'critical path'. Sometimes displayed as a Program Evaluation and Review Technique (PERT) chart, sometimes as a Gantt chart, the idea is to show the essential activities within a project, their earliest possible start and end times, their dependencies upon one another, and thus trace the shortest path between activities – the critical path.

Despite our familiarity with the diagrams hung on office walls or the terminology of events and lag time, we are still not very good at actually managing critical paths! This is, at least in part, because we have a tendency to forget why they are so valuable. A critical path allows us to see where many of the risks are – the dependencies and the delays – so that we can manage them. It is one of the most useful tools for a team to employ in their planning and along with the associated metrics can help us to measure our progress to improve predictability. A key point to make is that the critical path should be created by the team – it is a planning tool that belongs to them, not a contract to be handed down from on high. The team undoubtedly has greater knowledge about the actual length of activities, as well as the dependencies and order than someone divorced from the work itself.

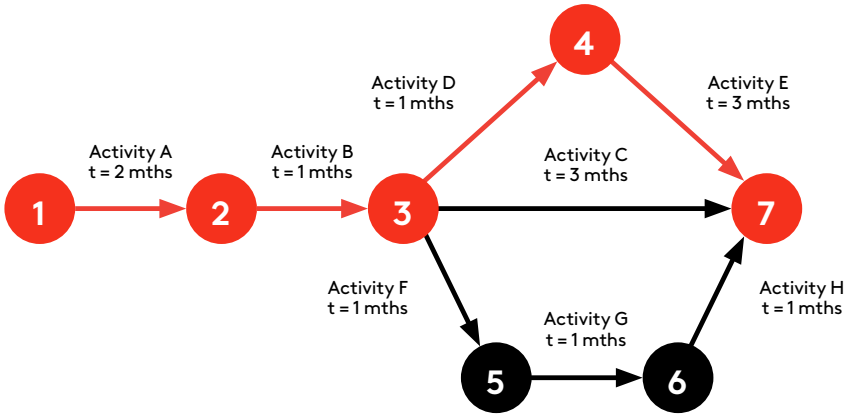


Figure 11. A PERT chart with critical path

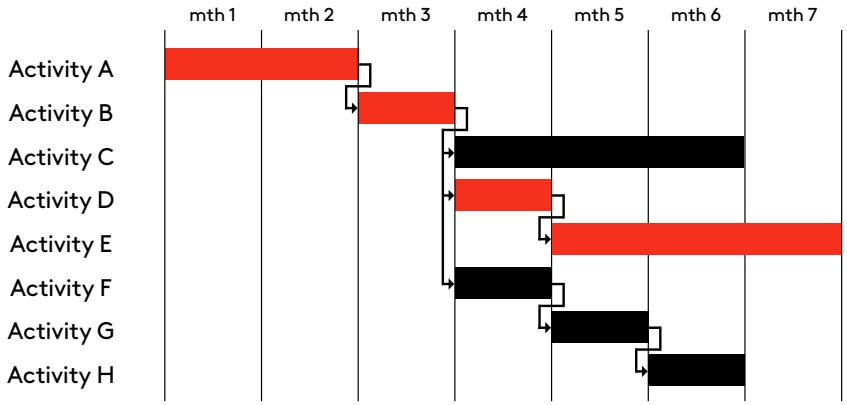


Figure 12. A Gantt chart with critical path

The PERT chart is for a project with seven milestones (1 through 7) and eight activities (A through H). The critical path is activities A, B, D and E (shown in red) giving the minimum project time of seven months with fast tracking. Activity C is sub-critical with a float of one month. Activities F, G and H are also sub-critical with a float of one month. The Gantt chart represents the same activities and critical path.

Sadly, the critical path has got rather a bad press – probably because it has often been misused. A critical path is neither a crystal ball showing us what will happen, nor an estimation guarantee to inform management when something will be delivered. Those Agile practitioners who treat 'milestones' or Gantt charts as if they were some kind of shameful habit are throwing away valuable tools. At the same time they can make little headway against ingrained, institutional desire for 'certainty'. The result is almost certain disaster and an organisation that announces 'we tried Agile – it didn't work'.

A critical path is so important because it shows us what can impact on our product delivery. If a critical activity slips by one day, then the product slips by one day. If a critical activity is delayed by one month, so is the product. It sounds blindingly obvious when stated like that, but one of the key outcomes should be to explicitly record assumptions – we assume this activity will take one week with a team of six people. If the team decreases in size for some reason, then the risk to the product delivery date should be clear to everyone. We can also examine other areas of likely risk – external dependencies, for example, or activities and events over which the team have no control.

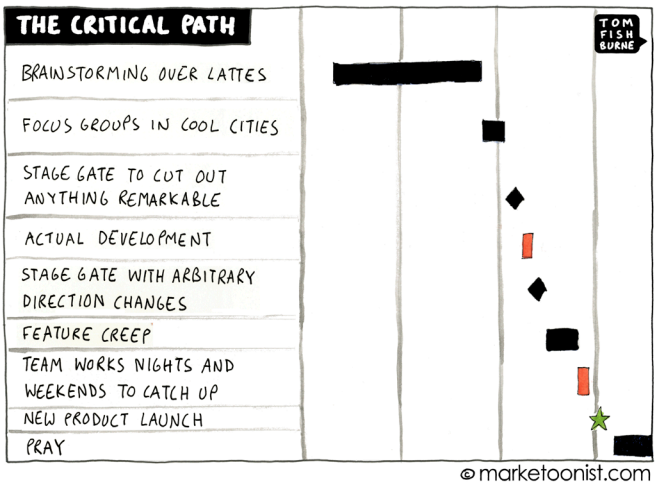


Figure 13. The critical path bogged down by inconsequential stuff

A critical path allows us to take our knowledge further and start making choices and decisions to mitigate the risks identified at these points. To do this intelligently we must be able to compare the value of delivering our product with the various costs that may be incurred. Such a framework demands a quantified cost of delay. We describe how to calculate a cost of delay in the Optimising Flow session. Here the point is that cost of delay is an essential planning tool, allowing the team to make economically valid choices in the organisation's best interests.

Some organisations actually reverse the true concept of the critical path. They calculate back from an idealised delivery date how late the product can be started using a minimum of resource (normally a budget calculation). To save money no resource is ever assigned to the team until absolutely necessary, while as soon as a specific task is done, the resource will be transferred elsewhere. Such thinking leads to waste in hand-offs, rework and task switching, as well as a less motivated and less productive team. By focusing on resource optimisation rather than speed of delivery such organisations leave themselves completely vulnerable to delays that could destroy the overall value. They have turned a useful planning tool into a means to increase risk, cost and delays!

There is a way to use these charts to great effect, and all it takes is a bit of consideration at each stage. The key concept behind critical path analysis is that you cannot start some activities until others are finished. The activities have to be completed in the right sequence and so it's essential that we keep updating the chart with what's actually happening, and re-schedule it according to real progress. Say some tasks are taking longer than expected – you need to decide whether they are going to continue at the same pace and if so, determine what impact this will have on the rest of the plan. The remaining effort now needs to be estimated and the project re-scheduled. Likewise, if tasks are ahead of schedule, the effect needs to be determined. In this case the impact on your resources will not be apparent just from looking at the charts – other tools will be required, such as resource histograms. These will identify peaks and troughs in resource utilisation. Tools and charts help us greatly in managing projects, but remember, they won't give us the answers.

Activity 5: Take a good look

This activity is a reflective one, although you may need to ask for help from your colleagues to gather the information.

Ask yourself the following questions. If you are unsure, involve others, but don't accept assurances – make sure you look for some data to back up the answer.

What level of slack or spare capacity do the team run with during a project? How low can the team keep their WIP?

Do they have authority to reject features not on the critical path?

Do senior managers appreciate the impact of approvals, funding decisions, etc. on the critical path? Are such delays tracked and reported on?

Are people assigned to a team early, even before their role is strictly 'necessary' for the tasks at hand?

Does the team know the cost associated with delays to the critical path?

Are they authorised to spend money to reduce delays within clear guidelines?

Commentary:

Often the answers to such questions are disappointing. In the worst-case scenario, delays at the 'fuzzy front-end' can eat into development time, but senior management blame the development team and the project manager for the subsequent delay, not the initial activity. Answering the questions above honestly can help show you and the organisation whether you are in danger of such a scenario. If so, it's best to confront the problem squarely. By following the guidance below with your own critical path and reporting honestly and transparently on delays and the activities that cause them, it enables the whole team to focus on the necessary changes.

2.1. Avoiding delays and managing dependencies

Most projects have activities that depend on other activities and many have dependencies on other projects, too. This creates a potential for delays which, as we know from a selection of other VFQ sessions, makes us slower. The higher the number of dependencies, the more likely it is that delays will happen. Here we consider the key techniques that planning should use to avoid or lessen the impact of such delays.

Overlap activities

The essential Agile view of critical path activities is that by overlapping them we can speed up the delivery. In a traditional waterfall critical path, all design might have to be done before any development could begin. In Agile, a small amount of design work feeds a small amount of development. This series of little cycles or small steps provides a different type of 'milestone', composed of iterations and increments.

Different elements of development can overlap. For example, before the final blueprint is finished, a supplier might receive approval to start tooling up. Naturally, this incurs some potential risk (the tooling might be wasted). The planner's job is to balance this risk with the benefit of moving faster (or to put it another way, with the cost and risk of going slower). Such decisions are why a cost of delay model is so important. Sadly it remains rarely used in project management, which is often more obsessed with cost management than maximising value.

Take activities off the critical path

We referred to Toyota earlier and this is a good time to come back to them. When the organisation began to optimise their manufacturing line, one of the brilliant insights was the gain to be made from taking activities off-line. A key example was the need to have sharp cutters. Previously the machine operator had taken a few minutes to sharpen these. By having two sets of tools – one in the machine and one being sharpened somewhere else – the team shaved a few precious minutes off the changeover time.

In product development we have to think about what activities can be taken off the critical path. What can be done in parallel? What can be done by someone else?

A classic example is to think about freeing up your critical resource. If you have a specialist who is proving a bottleneck, then what of their work can be taken away to free up their time? Can someone support them in doing the simpler elements of the job? Can someone do the finishing off or tidying up? There may be no need to hire a second specialist on the area, just to find a generalist who can manage the less complicated parts.

Bring uncertainty early

When an event has a large amount of uncertainty attached to it, wherever possible we want to take it off the critical path. If we don't know whether a specialist new technology will be ready in time then we want to either have a back up or ensure the activity is not on the critical path so that any delay will not delay the whole project.

If this is impossible, then we need to bring the uncertain activity into the critical path as early as possible. That way we will be warned of any overrun early on and can then manage it accordingly. Perhaps the best example for this is integration. It is a critical activity and if we leave it until the end of the project path then it has the capacity to make us horribly late. If we integrate early and often, then we can learn and manage our risk as we go. This is true of integrating software systems, software and hardware or external systems. Since any integration problems would be serious, we need to bring opportunities to test them together earlier into our path.

Bring the external in

This is a favourite adage of Agile, that the team should manage its external dependencies inside the team. In practical terms this means that specialist skills should be brought within the cross-functional team – this can range from marketing to technology expertise or manufacturing. It extends to include major external dependencies – a key supplier or customer, for example.

Having these people as part of the project team means there are less likely to be awkward surprises that imperil the critical path. Coordination between business processes should run more smoothly, while the team gains insight to and can prepare for likely problems. This works only if the people are genuinely part of the team – not just advisors who turn up to offer some criticism and then leave. Naturally any feedback from a customer or any catch-up meeting with a supplier is better than nothing, but the way to ensure a project goes really well is to have someone who is an active member of the team.

Bringing the outside in can also refer to ensuring the team has the appropriate authorities required – if sourcing some particular supply is a bottleneck then the team or team leader must have the authority to make decisions and sign-off invoices.

Companies often really struggle with the external elements because they want relationships to be formalised with contracts, non-disclosure agreements and orders, etc. defined in advance. Because this takes so long, the upfront work automatically drives big batches and increases the length of the critical path (all the negotiation work is holding up the project's speedy delivery).

Just as in the section about overlapping activities, companies must balance small up-front costs against cost of delay. When the risk feels too big or too uncertain, the answer is to try to reduce the size of the step needed. The whole order can't be signed off in advance, but perhaps an amount to assist with tooling can be; the total contract can't be agreed, but maybe the customer is happy to sign-off time and materials on three people beginning work on-site. Such agreements can keep the project moving and allow progress. In any case, the team will be learning more and can create an improved final contract, having already dealt with some of the uncertainty through actual experience.

The key question is always to ask – what small step can we take? If the customer is not prepared to pay for a single day of time, for example, then holding on and waiting for a one year contract is unlikely to create a successful partnership.

Focus on the part-time

In order to keep team sizes manageable and to share scarce resources, organisations often try to split individuals between several teams. Rather than creating full-time generalists that can turn their hand to whatever is next on the critical path, organisations are set up with numerous different experts. While understandable, a fragmented team is much slower and leads to bottlenecks that derail the critical path activities.

Don Reinertsen in his book Developing Products in Half the Time said that 'as a rough rule of thumb, assume that the less work someone has to do on your project, the more likely they are to be a delay on the critical path'. This may seem counter- intuitive to many who are used to putting the most responsibility and pressure on full-time team members, but it makes perfect sense when we think about it. The lower your involvement in a project, the less likely it is to be on your priority list. That means you are more likely to be late with your activity.

This is true not only for experts, but also for senior managers for whom 'approving' a decision on a project may be low on their personal priority list, but whose delays may cause significant project delays. It is also true when it comes to external suppliers or customers – where you are the major customer of a supplier, they are more likely to focus on you. When you are one of many, your small order will have a lower priority. Yet if the order is on your critical path, the effect could be disastrous.

What's not on the path

The critical path and the plan are not static. They change. This means that things which are not now on the critical path may well become so. This can be due to an activity expanding or being delayed, or receiving extra attention – the user interface and design, for example. Similarly, activities on the critical path might be moved off it if the team can complete them more quickly.

We need to be disciplined about keeping activities off the critical path, and not ignoring the work that is 'not yet critical'. No-one wants to be constantly panicking. As Don Reinertsen says, 'the critical path is the hot seat, it is not a pleasant place to be'.

Figure 14. No-one wants to sit in the hot seat

Kanban has a special class of service for 'intangibles', the activities that are not yet urgent but which must be done in case they become so. There's no point in not maintaining something until it breaks down, for example. Just as there's no point ignoring issues to do with users learning the new system just because they're not yet on the critical path.

Relax requirements

As deadlines approach, requirements get relaxed. Faced with not launching in time for a trade show or not having various help functions completed, managers normally accept that something may not be done for launch.

The problem is that this is rarely the best time to make such decisions. It may be that the help functions are more important to users than the back-end reporting that was completed months ago or supporting large numbers of browsers. Deciding in advance which are the essentials and which can be relaxed is an important part of trying to keep as much as possible off the critical path.

3 CONCLUSION

By now, we hope that the troublesome task of planning seems a little less daunting. We plan our nearby tasks in detail and organise our deliveries in to short iterations. Most importantly our plans should always be adaptable to change, to take into consideration the reality of the day as well as the general direction of travel. As with any aspect of a business, it comes down to our people. The people doing the work should be actively involved in building and developing the plan. They are the ones motivated to get it right and they have the skills to understand the day-to-day dependencies. With the strategy, portfolio and product plans, however, we need direction from a higher level to ensure that we have the right checks and balances in place to re-set expectations and align priorities.

We've discussed many techniques in this session and there are probably dozens more out there if you have a good look. Use this book as a reference, take each planning horizon and study the techniques then decide which ones best suit the scale and nature of your organisation. Or combine a few. There's no strict rule to planning. It's about testing out what works best for your business. As Richard Cushing said:

> "Always plan ahead. It wasn't raining when Noah built the ark."
> **Richard Cushing**

Learning outcomes

Now that you have completed this session, you will be able to:

Understand the multiple levels of planning in business

- Daily planning and stand-ups
- Iteration planning, backlog refining and commitment
- Release planning and feature selection
- Vision and strategic planning

Learn to apply planning tools and techniques

- Release plan and span plan
- The Business Model Canvas and the Lean Canvas

Develop existing theories to manage risk and investment through portfolio planning techniques

- Game theory and strategic decision making
- Real options, set-based design and deferring commitment

Formulate a strategy to deal with the issues of planning at scale with high dependencies

- Coordination, multi-level planning hierarchy breakdowns and combining multiple techniques

BIBLIOGRAPHY

Alleman, G., 2009. Plan The Work – Work The Plan. Herding Cats, [blog 11 September]. Available at: <http://herdingcats.typepad.com/my_weblog/2009/09/plan-the-work-work-the-plan.html>. [Accessed 12 November 2012].

Appelo, J., 2010. Management 3.0: Leading Agile Developers, Developing Agile Leaders. Addison Wesley.

Benson, J., 2011. Why Plans Fail: Cognitive Bias, Decision Making, and Your Business. Modus Cooperandi Press.

Cohn, M., 2005. Agile Estimating and Planning. Prentice Hall.

DeMarco, T., Lister, T., 2003. Waltzing With Bears: Managing Risk On Software Projects. Dorset House Publishing.

Fowler, M., 2005. The New Methodology, Predictive Versus Adaptive. [online] Available at: <http://martinfowler.com/articles/newMethodology.html#PredictiveVersusAdaptive>. [Accessed 24 October 2013].

Fowler, M., Taber, C., 2001. Planning And Running An XP Iteration. [online] Available at: <http://martinfowler.com/articles/planningXpIteration.html>. [Accessed 24 October 2013].

Gov.UK. What Does Agile Look Like?, Continuous Planning. [online] Available at: <https://www.gov.uk/service-manual/agile/what-agile-looks-like.html#continuous-planning>. [Accessed 24 October 2013].

Harvard Business Review, 2007. A Leader's Framework For Decision Making. [online] Available at: <http://hbr.org/2007/11/a-leaders-framework-for-decision-making/ar/1>. [Accessed 12 November 2012].

Harvard Business Review, 1984. How Senior Managers Think. [online] Available at: <http://hbr.org/1984/11/how-senior-managers-think/ar/1>. [Accessed 13 November 2012].

Harvard Business Review, 1997. Strategy Under Uncertainty. [online] Available at: <http://hbr.org/1997/11/strategy-under-uncertainty/ar/1>. [Accessed 9 November 2012].

Harvard Business Review, 2012. The True Measures Of Success. [online] Available at: <http://hbr.org/2012/10/the-true-measures-of-success/ar/1>. [Accessed 9 November 2012].

Harvard Business Review, 1999. Turning Goals Into Results: The Power of Catalytic Mechanisms. [online] Available at: <http://hbr.org/1999/07/turning-goals-into-results-the-power-of-catalytic-mechanisms/ar/1>. [Accessed 24 October 2013].

Highsmith, J., 2009. Agile Project Management: Creative Innovative Products. 2nd Edition. Addison Wesley.

InfoQ, 2013. Interview With Eduardo Miranda About Estimating and Planning Agile Projects. [online] Available at: <http://www.infoq.com/articles/eduardo-miranda-estimating-planning-agile>. [Accessed 24 October 2013].

InfoQ, 2011. The Many Levels of Planning on an Agile Project. [online] Available at: <http://www.infoq.com/articles/many-levels-planning-agile-project>. [Accessed 24 October 2013].

InfoQ, 2013. Martin Fowler at GOTO Amsterdam 2013 about Agile Essence and Fluency. [online] Available at: <http://www.infoq.com/news/2013/06/martin-fowler-agile-essence>. [Accessed 24 October 2013].

Kelly, A., 2011. Three Plans For Agile. [online] Available at: <http://www.allankelly.net/static/writing/webonly/ThreePlansForAgile_long.pdf>. [Accessed 24 October 2013].

Langr, J., Ottinger, T., 2011. Agile In A Flash: Speed-Learning Agile Software Development. Pragmatic Bookshelf.

Larman, C., 2008. Scaling Lean and Agile Development: Thinking and Organisational Tools for Large-Scale Scrum. Addison Wesley.

Liker, J,. 2004. The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer. McGraw-Hill Professional.

Marsh, J., 2008. Why Do We Plan Upfront? Signal vs. Noise [blog] 21 April, Available at: <http://37signals.com/svn/posts/982-why-do-we-plan-upfront>. [Accessed 17 October 2012].

McConnell, S., 2006. Software Estimation, Demistifying The Black Art. Microsoft Press.

Osterwalder, A., 2014. Value Proposition Design: How to Create Products and Services Customers Want. John Wiley and Sons.

Patton, J., 2014. User Story Mapping: Discover the Whole Story, Build the Right Product. O'Reilly Media.

Perkin, N., 2011. Agile Planning (Redux). Only Dead Fish. [blog 14 July] Available at: <http://neilperkin.typepad.com/only_dead_fish/2011/07/agile-planning-redux.html>. [Accessed 24 October 2013].

Pichler, R., 2013. Agile Product Planning: Vision, Strategy, and Tactics. Pichler Consulting [blog] 15 April, Available at: <http://www.romanpichler.com/blog/product-planning/agile-product-planning-vision-strategy-tactics>. [Accessed 24 October 2013].

Project Management Institute, 2009. A Guide To The Project Management Body Of Knowledge: PMBOK Guide. 4th Edition. Project Management Institute.

Ramusson, J., 2010. The Agile Samurai: How Agile Masters Deliver Great Software. Pragmatic Bookshelf.

Scaled Agile Framework. Release Planning. [online] Available at: <http://scaledagileframework.com/release-planning/>. [Accessed 24 October 2013].

Scaled Agile Framework, 2014. Program Portfolio Management. [online] Available at: <http://www.scaledagileframework.com/program-portfolio-management/>. [Accessed 4 March 2015].

Smith, P., Reinertsen, D., 1997. Developing Products In Half The Time: New Rules, New Tools. 2nd Edition. John Wiley & Sons.

Wheelwright, S., Clark, K., 1992. Revolutionizing Product Development: Quantum Leeps In Speed, Efficiency And Quality. Free Press.

Variety, 2015. Last Years Academy Awards Cost Over 40 Million. [online] Available at: <http://variety.com/2015/film/awards/last-years-academy-awards-cost-over-40-million-1201405003/>. [Accessed 5 March 2015].

Vasco, D., 2013. A Quick Trip To The Future. [online] Available at: <http://www.slideshare.net/duartevasco/a-quick-trip-to-the-future-land-of-no-estimates>. [Accessed 24 October 2013].

Yarow, J., 2014. This Is The International Grading System Google Uses For Its Employees – And You Should Use It Too. [online] Available at: <http://www.businessinsider.com/googles-ranking-system-okr-2014-1?IR=T>. [Accessed 4th March].

VFQ Value Flow Quality

valueflowquality.com

emergn.com

by emergn