Three plans for Agile

Allan Kelly, 2010

A brief guide to the three levels of forward looking plans used in Agile software development

One of the criticisms that has been levied at the Agile approach is that it foregoes planning. Indeed an observer with a traditional background might well notice the absence of plans. Yet if anything there is more planning in Agile than in traditional approaches, but the planning is different. It occurs not once at the start of a project but continually through the life of an Agile team.

The second difference between traditional planning and the Agile approach is who does it. Once upon a time planning was the preserve of the Project Manager (PM). Once the requirements had been written a PM would be assigned, he or she would create a plan - most likely using Microsoft Project to build a Gantt or Pert type chart. Project progress would then be measured against the plan.

In Agile, near time planning is a team exercise. Longer term plans may be created by one person but the aim of the plan is to garner input and support. The value is not so much in the plan itself as in the planning exercise. Team planning is a learning exercise for the whole team.

It is the near term plans - the iteration or sprint plans - that have received most of the attention in Agile. Yet these should not be the only plans. While no plan can accurately predict the future there is still a need for plans that facilitate co-ordination between teams. This is where *Release Plans* and *Roadmaps* come into play. Figure 1 shows how these three levels of plans nest. Iteration plans are the smallest and most detailed, while roadmaps are the largest and least detailed. In between lie site release plans.

Such plans can be used to plan further into the distance and also aid learning. These plans form the basis of discussions that allow more knowledge to be brought into play, priorities set and conflicts identified and resolved.

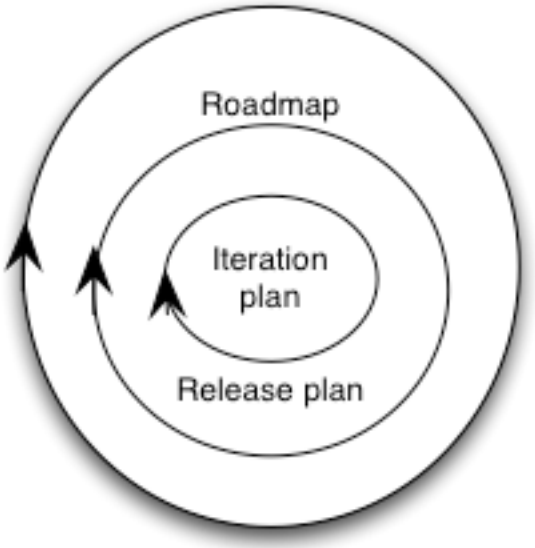

Figure 1 - Three plans fo r three time horizons

Iteration plans

At the start of each iteration - or sprint to use the Scrum term - the team set out what they will do in the next iteration. Since iterations typically last weeks - with two weeks being most common - these plans only peek a little into the future. They are however the most detailed plans, setting out exactly what the team will do.

The scheduling of work into an iteration plan is a discussion between the team members who will undertake the work and those who are charged with managing the requirements side of the project. Whoever this is, be they Product Owner, Customer, Product Manager, Business Analyst or Project Manager, proposes some work. Then using a velocity measure or a goal commitment protocol the team agree how much the team can accomplish. Once agreed the iteration plan is done and the team move to execution.

While the Product Owner nominates work to the plan it is agreed with the team through negotiation. Having the whole team involved in the planning removes the need to communicate the plan to the team, explain details and enthuse the team about someone else's plan. Once agreed the iteration plan is owned by the team and it is their collective responsibility to carry it out.

Since the plan only looks at the next few weeks it is unlikely to change much. Indeed, when it does change drastically it is usually a sign of problems. When this happens regularly it is a sign that action needs to be taken.

Release plans

For anything but the smallest piece of work it is likely some consideration needs to be given to a longer time frame than the next iteration. This is where release plans come in.

The simplest Agile model has teams releasing a new software version after each iteration. Perhaps a more common model has software releases after

several iterations - as shown in Figure 2. For example, a team might work on two-week iterations with quarterly releases - so four iterations makes one release. Release plans can help coordinate team routine and provide early warning about what is coming up.

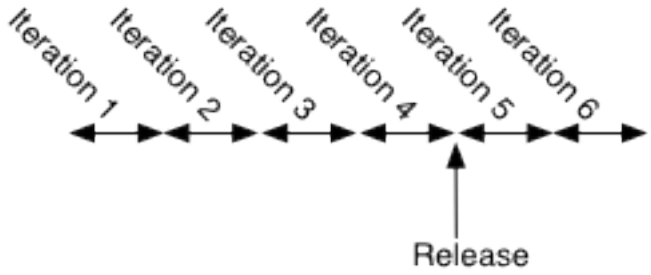

Figure 2 - Release every four iterations

Release plans look several iterations into the future, to the next release and for several releases after that. Looking forward a year is the absolute maximum for a release plan.

As always the further out the items on the plan the less detail and less certainty that is provided. While stories, features and functionality might be suggested for each release there is no certainty. The further away the date of a piece of work the more likely it is to change. This is simply a function of turbulent environment we work and live in.

Release plans are typically created and owned by the Product Owner based on information (e.g. velocity and estimates) and advice from the development team. The plans are regularly revised with the team to reflect passing iterations, changing requirements, priority adjustments and to add items from the roadmap in time.

Roadmaps

Roadmaps take up where Release plans leave off. Roadmaps look not weeks ahead but years ahead. Most likely they will be divided by quarter for the first year or two, beyond that by year. So a roadmap written in summer 2010 may well set out what is expected in each quarter of 2011, may sketch some ideas for each half 2012, and speculate what is in 2013 and 2014.

How far out they go depends on the nature of the product. For some products looking more than two years into the future may seem ambitious. In industries like telecoms future changes can be seen four, five or even more years ahead but details are unknown. For example, the world has known that 4G rollout will gather pace from 2010 onwards, but which technology, WiMax, LTE or some other, hasn't been clear.

Needless to say, roadmaps are highly speculative when it comes to product details and feature. But this is not the only information they need to contain. There is much relevant information that is known and can usefully be included on a product roadmap.

Roadmaps include not just projected product enhancements but, as far as is known, significant events in the future. Industry shows and changes to the law may be known several years in advance. Competitor product launches

and corporate fundraising events may also be included when known. Future changes in national and international standards, predicted technology changes and relevant socio-economic events need to be included whenever they have a bearing on the product and market.

Even when little is known about future product features it is possible to build a picture of the forces applying to the product. As time passes and distant events become near time it makes sense to transfer these events to the release plans. Once events start to register on iteration plans they may well be the entire focus of an iteration.

Roadmaps are created and owned by the Product Owner who is also responsible for showing them and incorporating feedback into the map. Product Owners sometimes keep roadmaps under wraps to prevent them falling into competitors hands or customers who believe they represent commitments. Development teams will benefit from understanding more about the roadmap and the future of the product.

To safeguard confidentiality Product Owners may choose to only communicate roadmaps in person or via physical documents. They may also create modified, or massaged, roadmaps for wider distribution.

Reversing the cone of uncertainty

The "cone of uncertainty" to describe the progress of projects. In the beginning the cone is wide open - as shown in Figure 3. As time passes projects reduce uncertainty as work is done, requirements understood, software developed and systems deployed. This description can be useful within projects and over the short to medium term, but, in the longer term, and for large systems the cone is actually reversed.

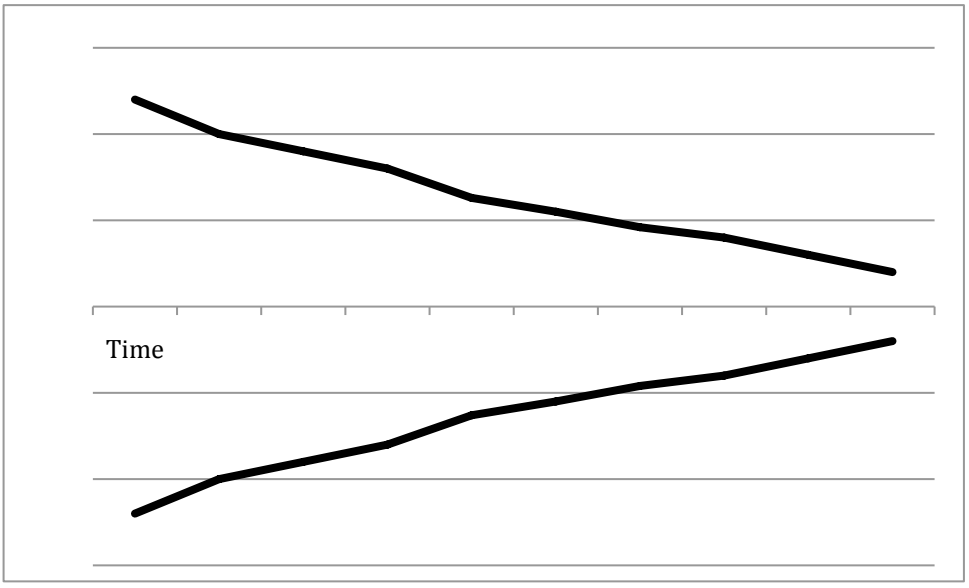

Figure 3 - Uncertainty decreases with time

We can be fairly certain about tomorrow: the weather, our bus fare and the time the bank will open. But next weeks weather is less certain, the bus fare next year will probably be higher but we don't know how much higher, and who knows if our bank will still exist in a year's time?

For a graphical illustration look at the GDP fan chart (Figure 4) produced by the Bank of England[1]. This and other charts of this type are used to forecast GDP and inflation several years into the future. Such forecasting bodies can predict next month's figures with reasonable accuracy, next year is less accurate and three years out less certain still.

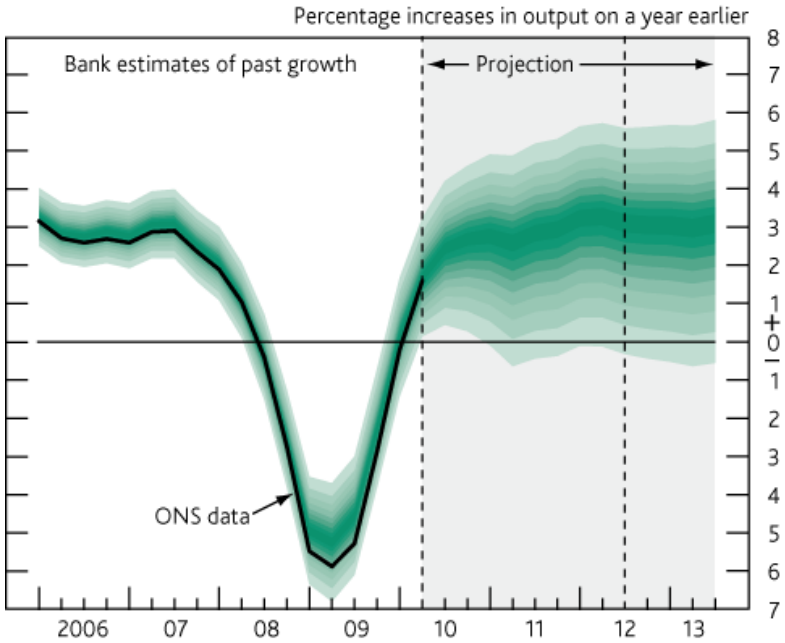

Figure 4 - Bank of England fan chart forecasting GDP from August 2010

The same is true for long-lived products and projects - and indeed the organizations for which we work. While we can determine requirements for the next iteration with little doubt we cannot forecast six months ahead. Who knows how well our product will be selling? How the competition will respond? Whether consumers will be buying, or whether the economy has taken a dip and destroyed our sales?

Predicting what our product will look like four years out is even more difficult. Indeed, our company may no longer exist - it may be sold, merged, bankrupt or in a new market.

Scenario planning

For many years military planners and major corporations have used a technique called scenario planning to try and prepare for the future. The oil company Royal Dutch/Shell was a notable user of scenario planning[2].

---

[1] See http://www.bankofengland.co.uk/publications/inflationreport/irfanch.htm, copyright retained by the Bank of England.

[2] See Schwartz 1991, *The art of the long view* for a full discussion on scenario planning.

The aim of scenario planning is not so much to forecast the future, rather it aims to help managers prepare for a range of possible solutions. To do this scenario planners incorporate know facts about the world (e.g. recent birth rates) and socio-economic forces (e.g. industrial productivity and marriage norms). Using this information the planners can paint a picture of future, possible worlds.

Explicit scenario planning can be used to add detail to release plans and roadmaps. For example, forecasting technology trends, customer behaviour to new products or Government reactions.

Release plans and roadmaps are a form of scenario planning used on Agile projects. Like traditional scenario plans, they exist not to mandate a future but educate the planners, help scare knowledge and facilitate learning through discussion.

All change

The key thing to remember about release plans and roadmaps is: they change. By default nothing on either plan represents a commitment. Of course if teams do make a commitment then it should be shown on the plan but most of the items on the plan describe an imagined world. The plans and descriptions exist to facilitate discussion and learning.

Plans are not fixed because the world is not fixed. A competitor may launch a new product which forces a response, a new technology may appear which necessitates a rethink in approach or even viability, the economy may turn down, or a major customer go bankrupt.

These plans need to be considered living documents. Only the iteration plan shows any certainty - and even that is questionable if the organization or Product Owner repeatedly changes goals while the iteration is in progress.

Finally

The promise of Agile has never been greater certainty about the future. Rather Agile promises more adaptability in an uncertain world. This isn't a failing of Agile, or Agile style planning, it is simply reality: Gantt and Pert charts never provided any certainty, only the illusion of certainty. Unfortunately, the appearance of certainty detracted from our adaptability and fooled customers into thinking the future is known.

Planning in Agile is different to traditional project planning. It happens on multiple levels, more often and involves more people. The aim is to promote learning and shared understanding.

The three type of plan are complementary, each related to a different time horizon. Each is changing, and each informs the other. Figure 5 summarises the plans.

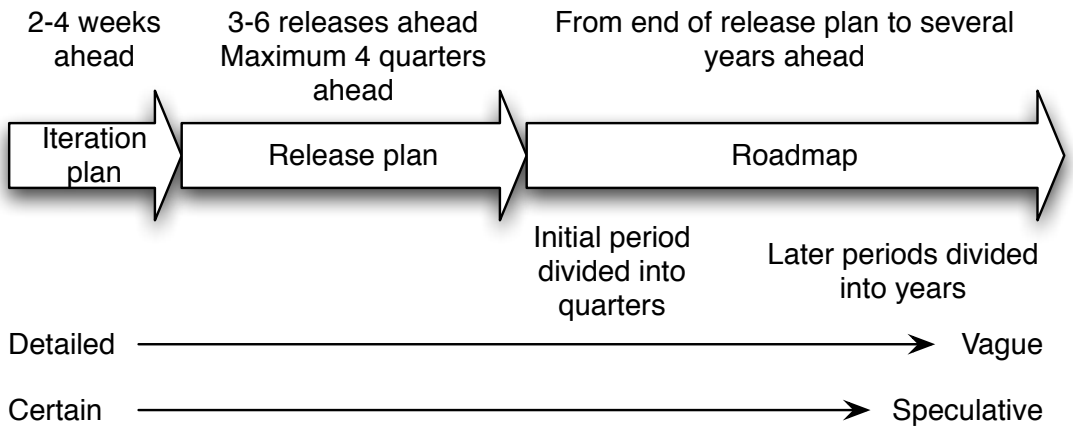

Figure 5 - Summary of plans

Only at the iteration level is there any degree of certainty as to what is being developed and when it will be ready. Consequently the only guarantees that can be given relate to the next few weeks.

Agile uses the planning process as a learning mechanism for the team, those who need to coordinate activity with other teams and commercial pressures. Plans serve to help image, visualise and learn about the future, not to coerce the future into some artificial state.

A shorter version of this piece was originally published on the Requirements Networking Group (RQNG) website (http://www.requirementsnetwork.com) in November 2010.

About the author

Allan Kelly has held just about every job in the software world: system admin, tester, developer, product manager and development manager. Today he provides training and coaching to teams and companies in the use of Agile and Lean techniques to develop better software with better processes.

He is the author of *Changing Software Development: Learning to become Agile*, numerous journal articles and is currently working on a book of *Business Strategy Patterns*.

More about Allan can be found at his website, http://www.allankelly.net. His blog is: http://blog.allankelly.net.