The topic is - Software Release Planning for Agile Project Management

**Definitions**

Benefits, challenges

Methods, tools, implementation

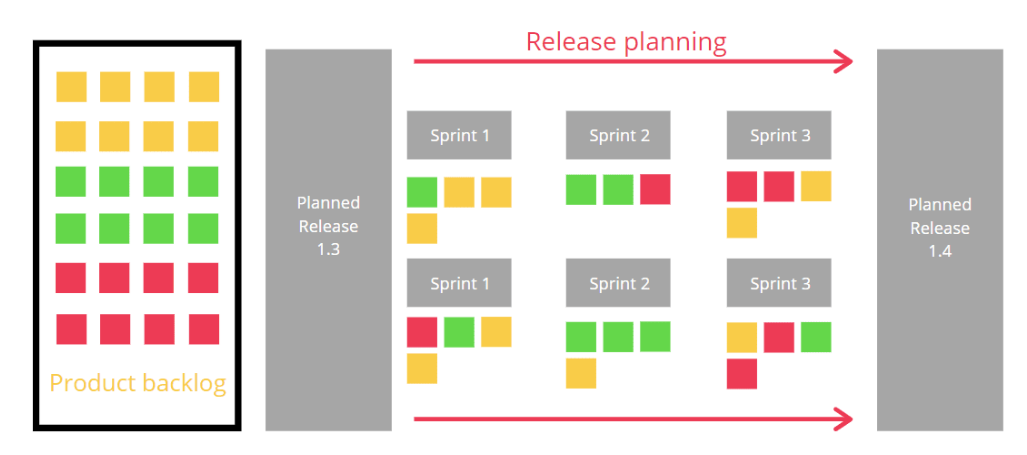
Case study/example

Conclusion

**What is Agile Release Planning:**

Agile release planning is a product management method where you plan incremental releases of a product. It differs from traditional software planning where you focus on major releases.

In Agile release planning, you prepare for staged releases and then break those down into several different sprints or iterations. Depending on your team structure and the size of the project, you may even have several sprints running at the same time.



**Why is agile release planning important?**

One reason is that businesses are increasingly going agile, and they need a release plan that works in that framework. A 2020 [McKinsey study](https://www.mckinsey.com/business-functions/people-and-organizational-performance/our-insights/enterprise-agility-buzz-or-business-impact) finds that businesses that go agile generally:

Increase operational performance by 30% to 50%

Increase employee engagement by 20% to 30%

Improve financial performance between 20% and 30%

**Purpose:**

A release plan outlines immediate future releases but doesn’t try to plan for years to come, but it also is unique from a product roadmap (high-level scope and timeline) because it goes into more detail. However, an Agile release plan doesn’t outline the work in each release. Instead, it batches iterations or sprints together into releases.

a release plan actually ensures that you create a coherent version of your product every time. It’s a great tool for combining changes that will have a significant impact on the user experience in a shorter period of time.

Release planning (RP) addresses decisions related to selecting and assigning features to create a sequence of consecutive product releases that satisfies important technical, resource, budget, and risk constraints. A good release plan should

* provide maximum business value by offering the best possible blend of features in the right sequence of releases,
* satisfy the most important stakeholders involved,
* be feasible with available resources, and
* reflect existing dependencies between features.

**Steps of Release/Iteration planning process: (We can represent these steps in a flow diagram)**

The release planning process includes the following activities:

The release planning game typically involves estimating 2–3 months worth of stories. The estimates are important, as they are the basis for planning the next release in terms of prioritising features and staffing the development team. In some situations, the estimates are also used for contract negotiations and bidding.

* Features are broken down into user stories through close communication between development team and customer.
* The user stories are identified for the release.
* The development team estimates the size of each user story.
* The product owner prioritizes the user stories.
* The product owner identifies the release features given the available resources.
* The product manager and the development team estimate the release date.
* The product owner gets the development team commitment on the release date.

The iteration planning process includes the following activities:

In the iteration planning, user stories are broken up into development tasks. Responsibility for each task is assigned to a team member. Estimates at this level are detailed, but as weekly or biweekly iterations are most common, the number of stories involved in each planning session is limited. In addition to providing new estimates for the user stories that can be used for more detailed planning, the re-estimation process provides a setting for the team to discuss design decisions and implementation strategies.

* The product owner and the development team establish an iteration goal.
* The team members estimate the work that they will be able to complete in the iteration.
* The development team chooses the user stories from the product backlog that support the iteration goal.
* The development team confirms that the selected user stories can complete the goal planned for the iteration.
* If the development team finds any of the user stories do not fit in the current iteration, they can remove them from the iteration.
* The development team creates the iteration backlog tasks associated with each user story.
* Team members break the user stories into individual tasks.
* Team members allocate a number of hours to each task.
* The development team confirms that they can complete the tasks in the time available in the iteration.
* If a task exceeds the hours available, the development team will discuss with product owner what tasks/user stories are the best to remove.
* If the development team finds an extra time available within the iteration, they can include another user story.
* Each team member selects his/her tasks to accomplish.

Methods:

**Planning poker**

With the planning poker estimation process, each developer was given index cards with the numbers 0.5, 1, 2, 3, 4 and >4 written on them. After the initial discussion that lasted until the developers felt they knew enough about the user story to estimate the required effort, all developers would estimate individually and simultaneously show the card corresponding to their estimated number of pair days for this story.

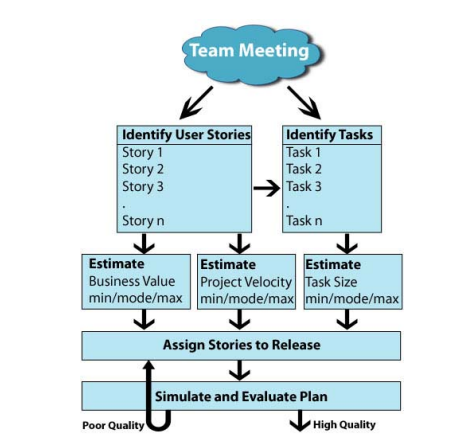
If the estimates differed, the developers with the lowest and highest estimate would justify their estimate. A short discussion would follow, where the developers sought to agree on a final estimate for the task. Usually the developers would reach agreement fairly quickly. In the few cases where they did not, the project manager, playing the role of moderator, would make a decision.

### Planning poker is more accurate when the team has previous experience from similar tasks

### Planning poker is possibly less accurate when there is no previous experience from similar tasks

Possible explanations to why planning poker fails in this respect, is:

1. If the developers happen to suggest similar estimates for a story where they have no prior experience, they might feel a false sense of security that the estimate is reasonable.
2. The opposite could also be true, that the developers become more cautious, taking the various eventualities mentioned by all developers into account, and therefore overestimating the complexity of the task.

****

**Features that tools provide for release planning:**

* Create New Release
* Edit/Delete User Story
* Release Burn-Down Chart
* Create New Iteration
* Calculate the Iteration Velocity
* Iteration Burn-Down Chart

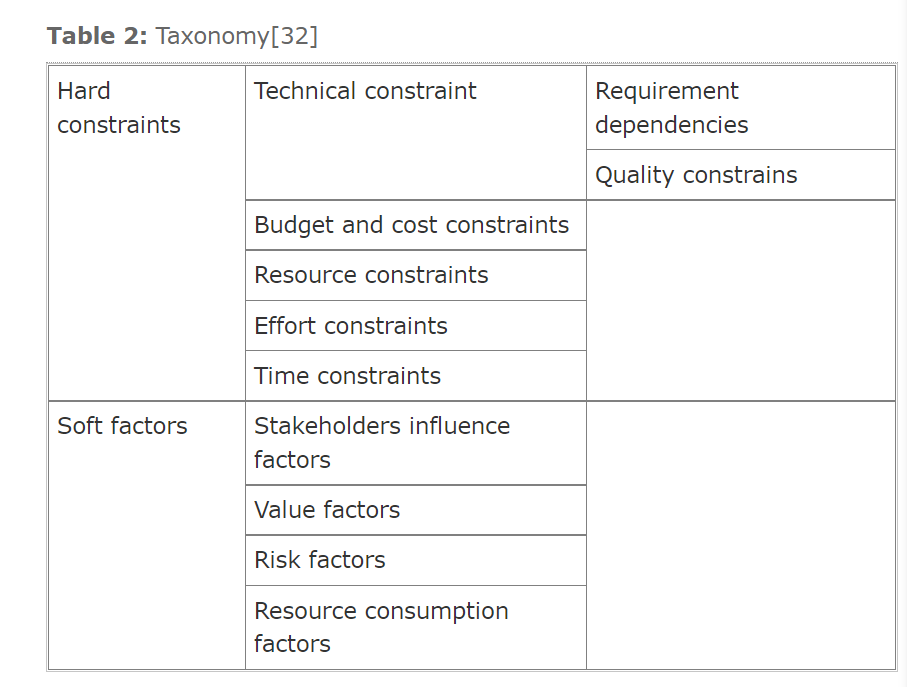
**Evaluation**

The release planning problem is characterized by various constraints. The technological constraints are considered where features are either in a coupling relation such that they require being included together in a release or they are in a precedence relation such that a feature needs to be implemented before another. There may be constraints in giving priorities to the stakeholdersEach stakeholder is assigned a weight between (0,1) such that the stakeholder weights are normalized to 1 Each stakeholder p assigns an ordinal value, value (p,i) to each feature i based on the importance of the feature to the specified stakeholder. Each feature consumes different types of resources such as effort; budget etc. for its implementation. Every release will have a certain amount of resource capacity of type t available for it. Thus, the amount of resources allocated to features in a release should not exceed the resource limit for the release.

**A comparison of model-based and judgment-based release planning**

A formal release planning model purports to make explicit and prescribe the data and the analyses needed to generate well-founded release plan alternatives. Tools based on such models are then designed to aid in various stages of a more formal release planning process according to a given model.

judgment-based planning (referred to as *ad hoc* planning in Svahnberg et al's review), is based on human judgment and assumes that a project is able to handle a sufficient amount of release planning concerns and tradeoffs through project members' mental processes and through more or less informal negotiations between stakeholders.

****

References:

<https://www.teamwork.com/blog/release-planning/>

<https://ieeexplore.ieee.org/document/7516149>

<https://ieeexplore.ieee.org/abstract/document/1524914>

<https://ieeexplore.ieee.org/document/1667560>

[**https://ieeexplore-ieee-org.ezproxy.gl.iit.edu/stamp/stamp.jsp?tp=&arnumber=4492425&tag=1**](https://ieeexplore-ieee-org.ezproxy.gl.iit.edu/stamp/stamp.jsp?tp=&arnumber=4492425&tag=1)

[**https://ieeexplore.ieee.org/document/5676859**](https://ieeexplore.ieee.org/document/5676859)

[**https://ieeexplore.ieee.org/document/6032518**](https://ieeexplore.ieee.org/document/6032518)

[**https://www.huffpost.com/entry/why-agile-release-plannin\_b\_10113514**](https://www.huffpost.com/entry/why-agile-release-plannin_b_10113514)