BITS Pilani

Pilani Campus

BITS Pilani presentation

K.Anantharaman
kanantharaman@wilp.bits-pilani.ac.in

innovate achieve lead

BITS Pilani

Pilani Campus

Module-6 Agile Planning & Release Planning

Agile Planning

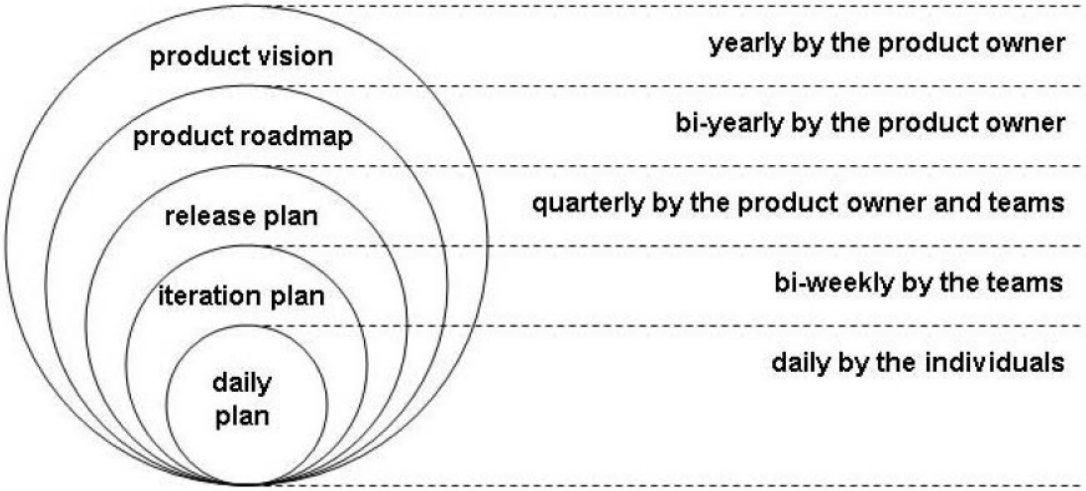
MANAGING ORGANIZATIONS

Bring Agile Planning to the Whole Organization

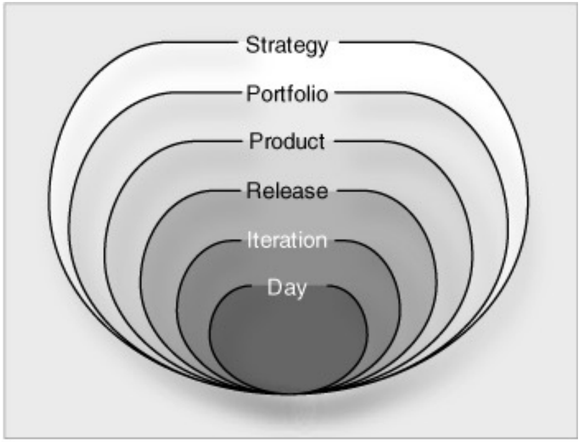
by Jeff Gothelf

https://hbr.org/webinar/2015/05/bring-agile-planning-to-the-whole-organization

Agile Planning



Single Product Organization



Many Products or Services Organization

An Enterprise Agile Framework

Ref: 5 Levels of Agile Planning: From Enterprise Product Vision to Team Stand-up by Hubert Smits

Release Planning

Inputs:

Product Vision

Product Road Map

Product Backlog

Release Backlog, Velocity, Iteration length, Trade off-matrix (Scope, Cost, Schedule)

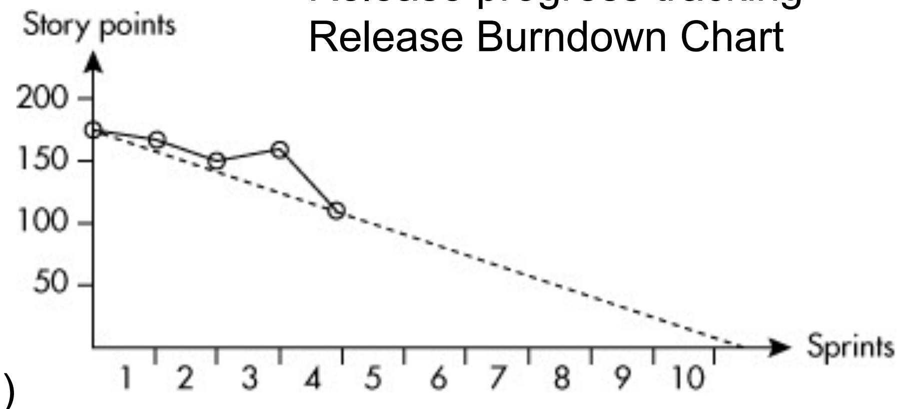
Release Planning
Sprint 1
Sprint 2
Sprint 3
R1
R2
R3

Example: Release planning

Inputs:

- Release backlog - 50 User stories
    - (200 Story points)
- Velocity = 20 Story points
- Iteration Length - 2 weeks
- Budget = $200,000
- Cost of each Iteration = $20000
- Trade of Matrix :
- Schedule (Fixed), Cost(Fixed), Scope(Flexible)

Release progress tracking
Release Burndown Chart



Outputs:

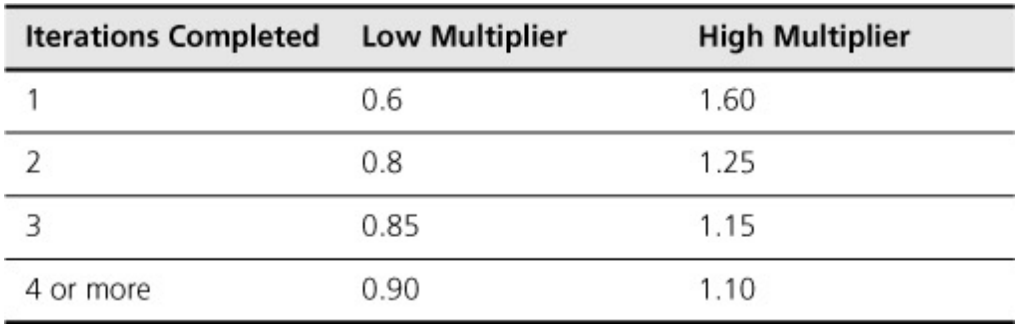
- Total number of Iterations required = 10 Iterations (200/20)
- If you are planning for 2 releases
- Number of iterations per release = 5 Iterations
- Duration of each release = 5*2 = 10 weeks

- Suppose, Iteration cost = $25000; only 8 iterations is possible.
- 160 points can be delivered (Scope may have to be reduced)

Source: Agile Estimation and Planning by Mike Cohen

Estimating Velocity

- Use historical values.
- Run an iteration
- Make a forecast.
- You should consider expressing the estimate as a range.
  - Example: If your team velocity is 20 story points - You have a very limited chance of being correct in future. Instead give a range 15-24 story points



Iterations Completed	Low Multiplier	High Multiplier
1	0.6	1.60
2	0.8	1.25
3	0.85	1.15
4 or more	0.90	1.10

Source: Agile Estimation and Planning by Mike Cohen

Source: Agile Estimation and Planning by Mike Cohen

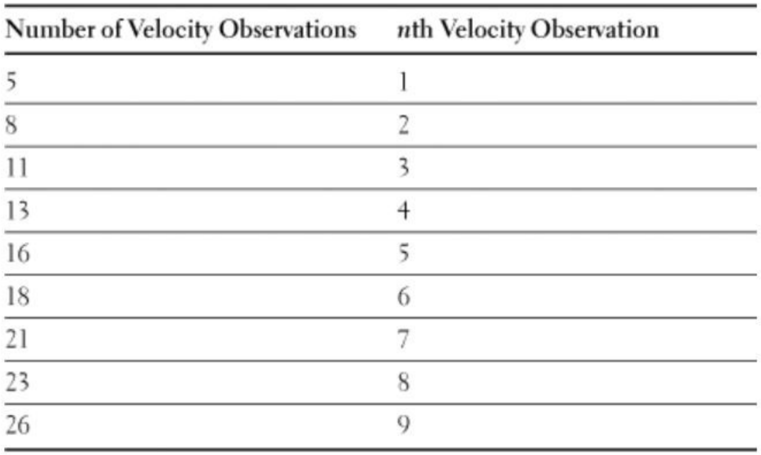
High-confidence forecast- Example

- Velocity of completed **8 sprints**
- :20, 25, 28, 26, 16, 20, 26, 26

Sorted Velocity List

16
20
20
25
26
26
26
28

90% Confidence Interval



Number of Velocity Observations	*n*th Velocity Observation
5	1
8	2
11	3
13	4
16	5
18	6
21	7
23	8
26	9

20 ➔ Lower confidence, certainly we will do
23 ➔ Mean Velocity – We will get here
26 ➔ Upper confidence, Most we could expec

Use the nth Lowest and the nth Highest Observation of a Sorted List of Velocities to Find a 90% Confidence Interval

Source: Succeeding with Agile SW development by Mike Cohen

Creating a Release Plan Exercise

- The backlog for this release has 140 story points, with a start date of D0 and a sprint length of 2 weeks. Range of estimated velocities: Low = 18; High = 20
- The average velocity of the first two sprints was measured to be 15 Story Points.

1. Calculate the maximum and minimum schedules, as well as the points that can be completed per sprint, by maintaining the same velocity range.

2. What is the maximum and minimum timeline and number of points that can be completed if the budget is $140000 and the cost of a sprint is $20000?

Creating a Release Plan Exercise

1. Calculate the maximum and minimum schedules, as well as the points that can be completed per sprint, by maintaining the same velocity range.

- Velocity High =20; Number of Iteration Required = 140/20 = 7
- Number of Story points completed in first two sprints= 30
- Remaining Story points = 110
- Number of iteration required to complete the 110 points = 110/20 = 5.5 ~ 6 Sprints
- Sprint 1-2 = 15 points; Sprint 3-7 = 20; Total number of points that ca be delivered = 130

- Velocity low = 18; Number of Iteration Required = 140/18 ~ 8 Sprints
- Number of Story points completed in first two sprints= 30
- Remaining Story points = 110
- Number of iteration required to complete the 110 points = 110/18 = 6.1 ~ 7 Sprints
- Sprint 1-2 = 15 points; Sprint 3-7 = 18; Total number of points that ca be delivered = 120

Creating a Release Plan Exercise

2. What is the maximum and minimum timeline and number of points that can be completed if the budget is $140000 and the cost of a sprint is $20000?

- Available budget is $140000, Each Iteration cost = $20000; Only 7 Iterations is possible.
- Max and Min Schedule is same = D0 +14 Weeks
- Velocity High =20; Number of Iteration Required = 140/20 = 7
- Number of Story points completed in first two sprints= 30
- Remaining Story points = 110
- Number of iteration required to complete the 110 points = 110/20 ~6

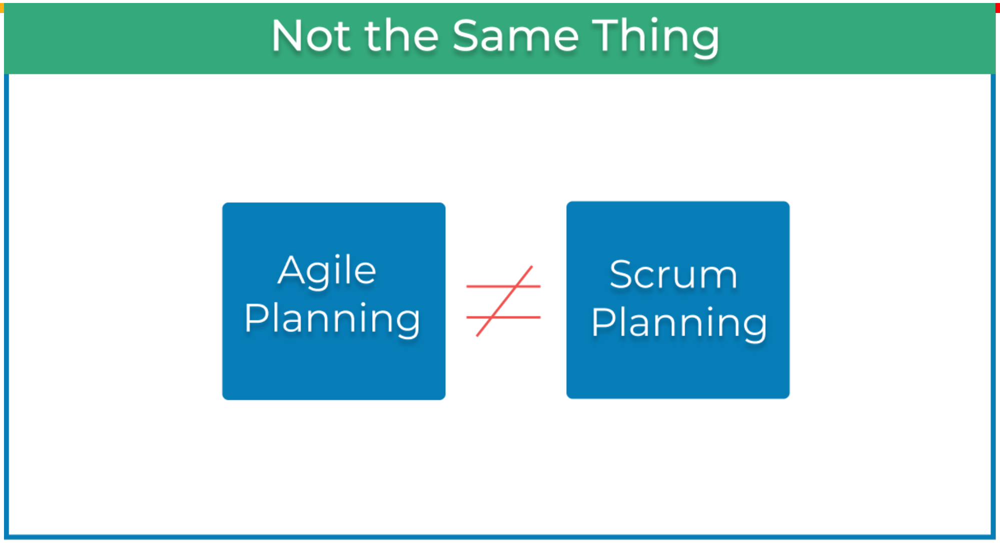
- Sprint 1-2 = 15 points; Sprint 3-7 = 20 points ;
- Total number points that can be delivered = 130 points

- Velocity low = 18; Number of Iteration Required = 140/18 ~ 8 Sprints
- Number of Story points completed in first two sprints= 30
- Remaining Story points = 110
- Number of iteration required to complete the 110 points = 110/18 ~ 7
- Max Schedule = D0+ 14 weeks

BITS Pilani
Pilani Campus

Module-6 Agile Planning & Release Planning – Additional Notes

Agile Planning



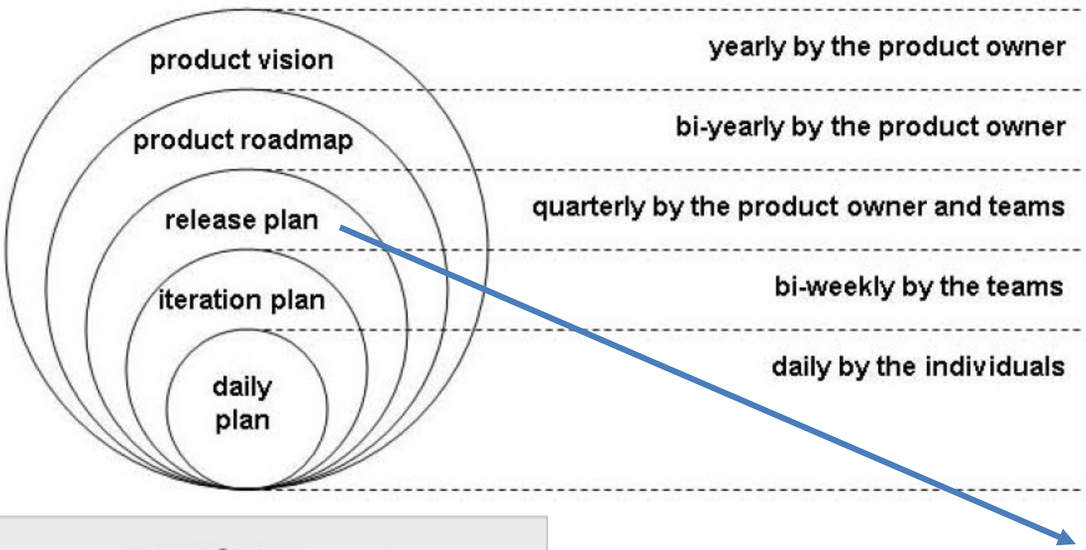
Not the Same Thing

Agile Planning ≠ Scrum Planning

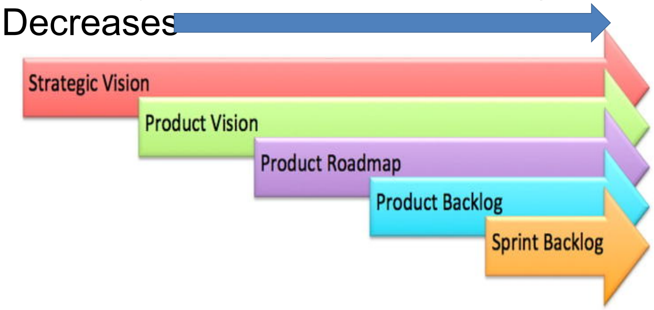
- Agile thinking applied across various industries and not just software.
- It is more important to know how to apply generic techniques and practices on the global company level, irrespective of the type of business.

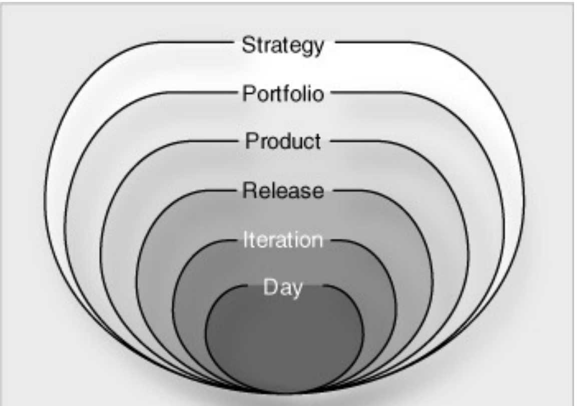
Source : https://kanbanize.com/agile/project-management/planning

5-Levels of Agile Planning (Product Planning)



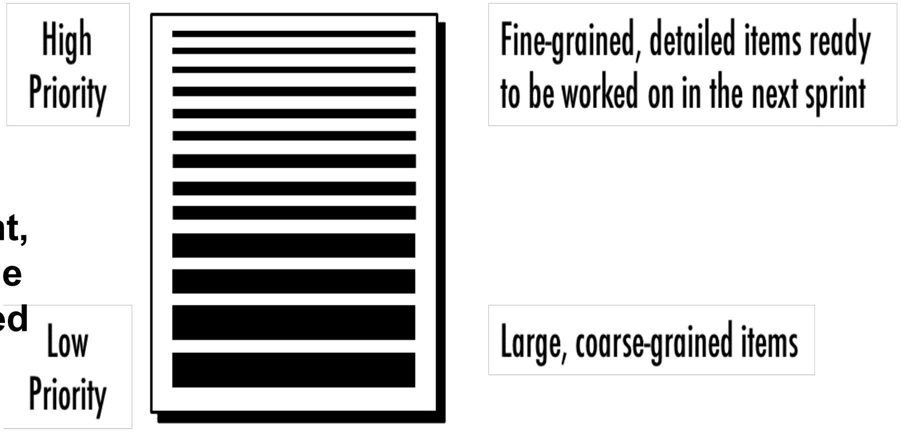
Flexibility to accommodate change Decreases





An Enterprise Agile Framework

DEEP:
Detailed,
Emergent,
Estimable
Prioritized


High Priority
Low Priority
Fine-grained, detailed items ready to be worked on in the next sprint
Large, coarse-grained items

Ref: 5 Levels of Agile Planning: From Enterprise Product Vision to Team Stand-up by Hubert Smits

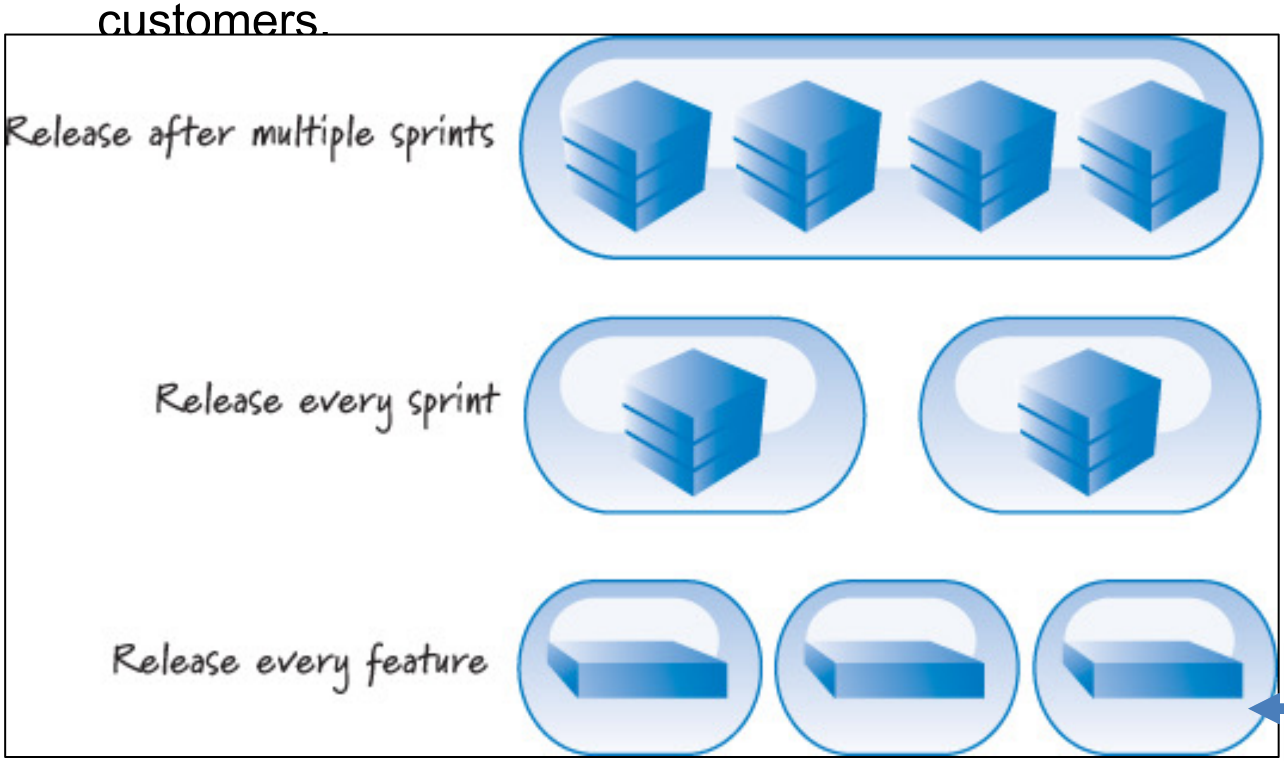
5-Levels of Agile Planning (Product Planning)

- Each of the five levels of planning addresses the fundamental planning principles: priorities, estimates and commitments.
- 5 Levels of Agile Planning is aimed to avoid big upfront design
- Most agile teams are concerned only with the three innermost levels (Day, Iteration, Release) of the **planning onion**.
- Involve stakeholders in planning, Review the plans frequently

5 Levels of Agile Planning: From Enterprise Product Vision to Team Stand-up by Hubert Smits

Patterns of Release Planning/Different release cadences

- Many organizations have its own cadence regarding release of products to its customers.



Continuous deployment (or Continuous delivery)

Whichever release cadence being followed, Most organizations find some amount of longer-term, higher-level planning to be useful. We refer to this type of planning as release planning

Ref: Essential Scrum: A Practical Guide to the Most Popular Agile Process by Kenneth S. Rubin Published by Addison-Wesley Professional, 2012

Agile Release Planning

- Release planning is an important task for product people working with agile teams:
  - It ensures that the product is moving in the right direction and it connects Product strategy and tactics.
- Release as a version of a product:
  - For example, Mac OS X Catalina and Windows 10.
  - Releases come in two flavors: major releases, like iOS 13, and minor releases, such as iOS 13.3.
- Release planning is the process of determining the desired outcome of one or more major releases and maximizing the chances of achieving it.

Source: https://www.romanpichler.com/blog/release-planning-advice/

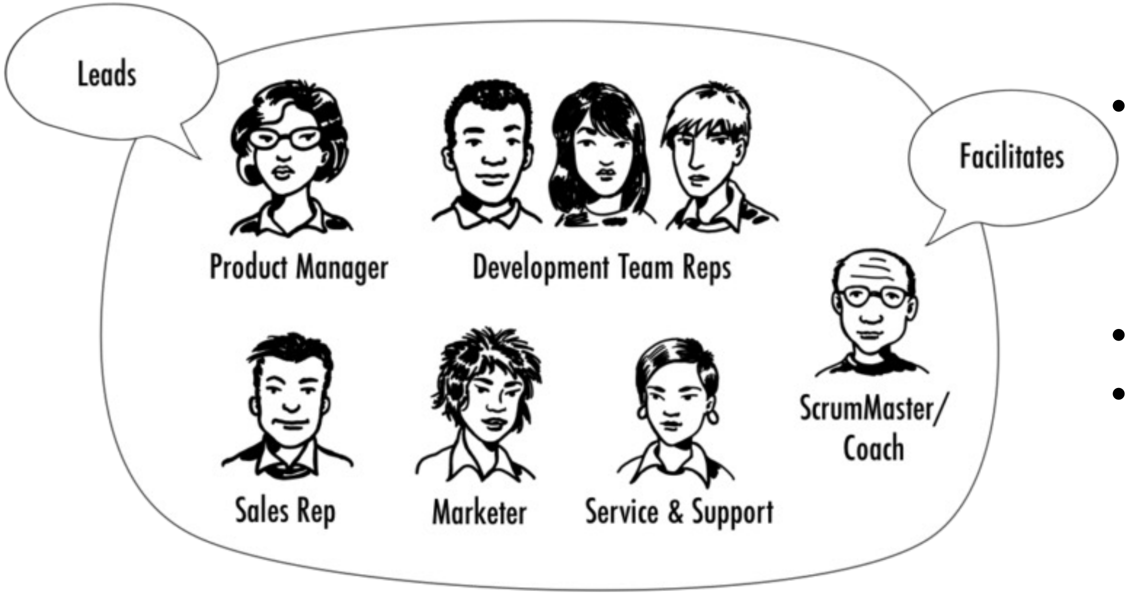
Agile Release Planning …

- Agile release planning provides a high-level summary timeline of the release schedule (typically 3 to 6 months).
- Agile release planning also determines the number of iterations or sprints in the release.
- Allows the product owner and team to decide how much needs to be developed and how long it will take to have a releasable product based on business goals, dependencies, and impediments.

Source: https://www.romanpichler.com/blog/release-planning-advice/

Make Release Planning Collaborative

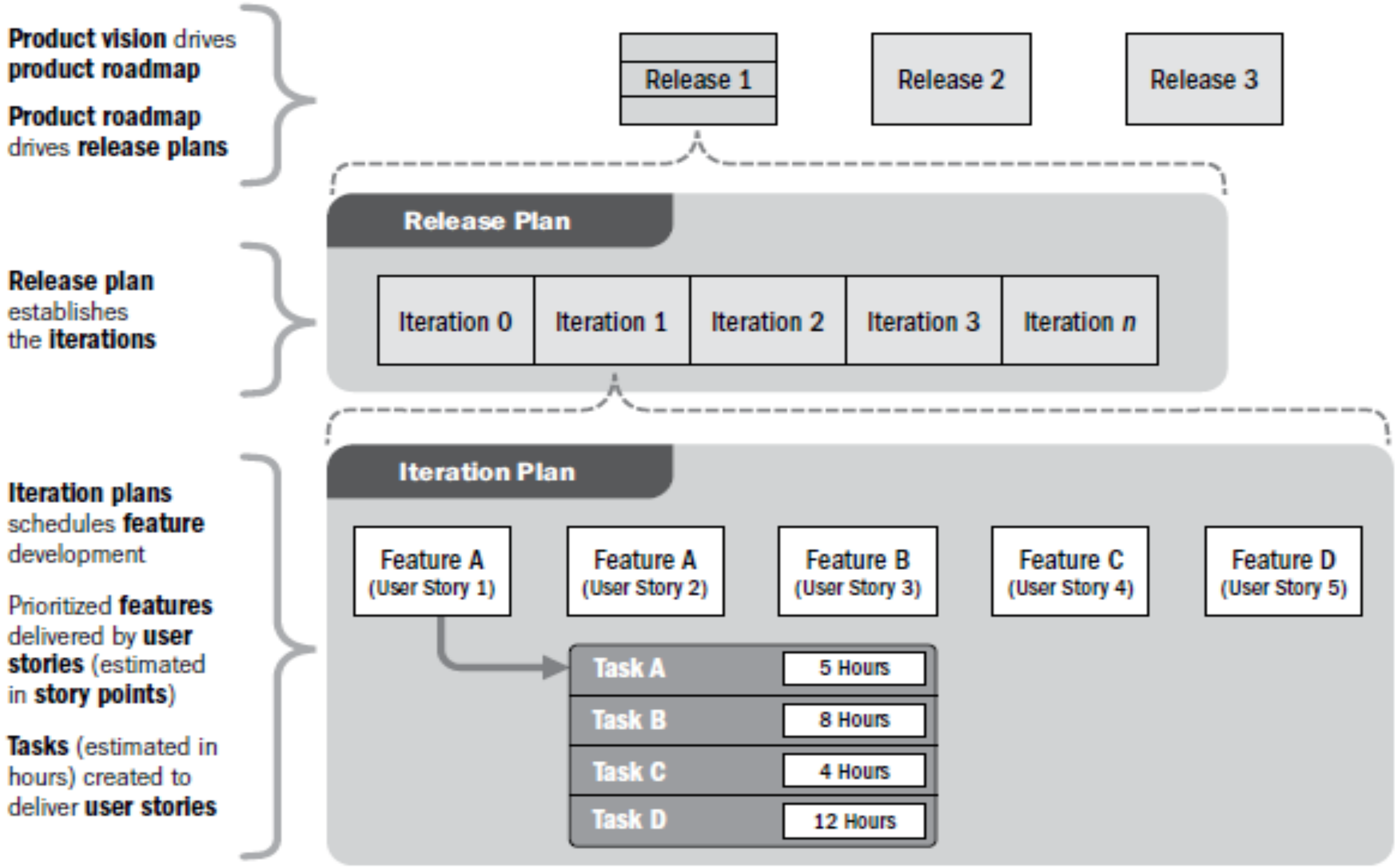
- Release planning is best done as a collaborative effort by involving the stakeholders and the development team



- Schedule regular roadmapping sessions.
- Possibly as part of your strategy review process and invite key stakeholders and development team members.
- Discuss Release Progress
- Invite Stakeholders to Sprint Review meetings.

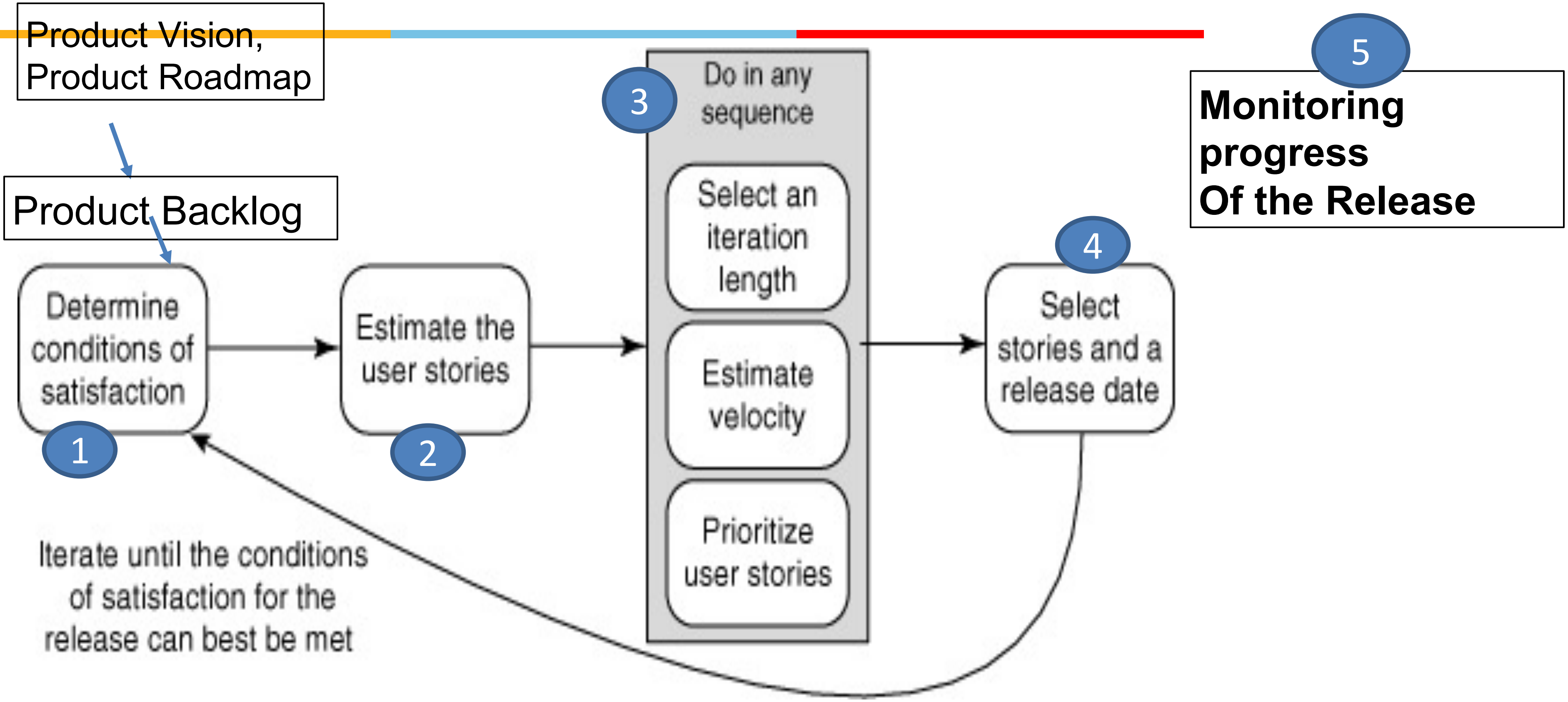
Source: https://www.romanpichler.com/blog/release-planning-advice/

Relationship between product vision, product roadmap, release planning, and iteration planning.


Product vision drives product roadmap
Product roadmap drives release plans
Release plan establishes the iterations
Iteration plans schedules feature development
Prioritized features delivered by user stories (estimated in story points)
Tasks (estimated in hours) created to deliver user stories
Release 1
Release 2
Release 3
Release Plan
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration n
Iteration Plan
Feature A (User Story 1)
Feature A (User Story 2)
Feature B (User Story 3)
Feature C (User Story 4)
Feature D (User Story 5)
Task A
5 Hours
Task B
8 Hours
Task C
4 Hours
Task D
12 Hours

Source: PMI.ORG

The steps in planning a release.


Product Vision,
Product Roadmap
Product Backlog
Determine conditions of satisfaction
1
Estimate the user stories
2
3
Do in any sequence
Select an iteration length
Estimate velocity
Prioritize user stories
4
Select stories and a release date
5
Monitoring progress
Of the Release
Iterate until the conditions of satisfaction for the release can best be met

1 Use Trade-off Matrix (Fixed, Flexible, Accept), Date, Scope, Cost - Fix important factor

Given a fixed **schedule**, we will choose a level of **resources** and adjust the **features** set as necessary.

Source: Agile Estimation and Planning by Mike Cohen

Development Constraint Combinations

Project Type	Scope	Date	Cost
Fixed Everything (Not Recommended)	Fixed	Fixed	Fixed
Fixed Scope and Date (Not Recommended)	Fixed	Fixed	Flexible/Accept
Fixed Scope	Fixed	Flexible	Fixed/Flexible
Fixed Date	Flexible	Fixed	Fixed

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Condition of satisfaction

- **Establishing clear, specific, and measurable goals.** Call these goals product or release goals - captured in product roadmap.
- **Prioritize the Success Factors for Releases:**
  - But in reality, unforeseen things do happen. The development progress may not be as fast as anticipated, for instance, or one of the technologies may not work as expected.
  - Use Trade-off Matrix (Fixed, Flexible, Accept)
  - Date, Scope, Cost - Fix important factor
- **Quality:** Quality should be fixed and not be compromised. Otherwise, responding to user feedback and changing market conditions and quickly adapting your product will be hard, if not impossible.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Estimate User Stories

- It is not necessary to estimate everything that a product owner may ever want.
- It is necessary only to have an estimate for each new feature that has some reasonable possibility of being selected for inclusion in the upcoming release.
- Often, a product owner will have a wish list that extends two, three, or more releases into the future. It is not necessary to have estimates on the more distant work.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Factors in Select an Iteration Length

- The length of the release being worked on
- The amount of uncertainty
- The ease of getting feedback
- How long priorities can remain unchanged
- Willingness to go without outside feedback
- The overhead of iterating
- How soon a feeling of urgency is established
- Make a Decision and stick to the Rhythm
- 2 weeks sprint is ideal.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

The Overall Length of the Release

- Short projects benefit from short iterations.

The length of a project's iterations determines:

1. How often the software can be shown and progress measured?
2. How often the product owner and team can refine their course, because priorities and plans are adjusted between iterations.
   - Opportunities to gather end-of-iteration feedback
   - General rule of thump: Aim for five to six feedback opportunities per release.
   - Example: 3 months release
   - Iteration length : 2 weeks – 5 times feedback for course corrections-ok
   - 4 weeks iteration provides only two times feedback- not ok

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

The Amount of Uncertainty

Uncertainty comes in multiple forms.

- User need
- Technical aspects
- Team Velocity

- The more uncertainty of any type there is, the shorter the iterations should be.
  - Shorter terations allow more frequent opportunities for the team to measure its progress through its velocity and more opportunities to get feedback from stakeholders, customers, and users.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

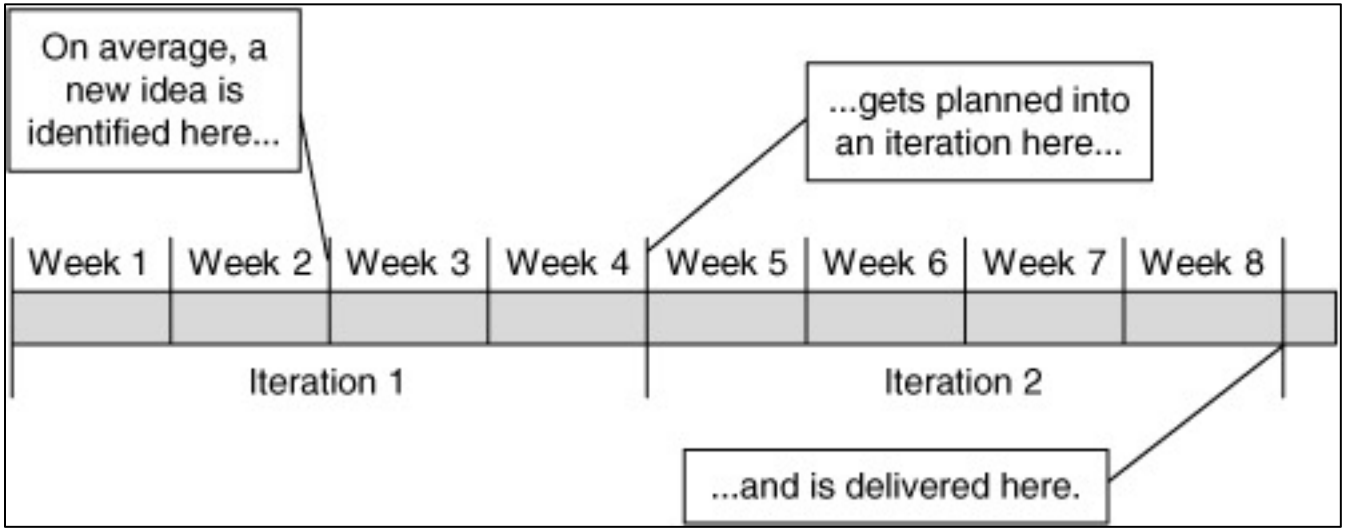
The ease of getting feedback

- Choose your iteration length to maximize the value of the feedback that can be received from those inside and outside the organization.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

How Long Priorities Can Remain Unchanged

- Once a development team commits to completing a specific set of features in an iteration, it is important that that the **product owner not change priorities** during the iteration, also protect the team from others to change the priorities.


On average, a new idea is identified here...
...gets planned into an iteration here...
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Iteration 1
Iteration 2
...and is delivered here.

- It takes an average of 1½ iterations to go from idea to software.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Willingness to Go without Outside Feedback

- Even with a well-intentioned and highly communicative team, it is possible that the results of an iteration could be found worthless when shown to the broader organization or external users at the conclusion of the iteration.
  - This may happen if the developers misunderstand the product owner (and don’t communicate often enough during the iteration).
  - It could also happen if the product owner misunderstands the needs of the market or users.
- Less often a team receives outside feedback, the more likely we are to go astray and the greater the loss will be when that happens.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

The Overhead of Iterating

- There are costs associated with each iteration.
- For example, each iteration must be fully regression tested:
  - If this is costly (usually in terms of time), the team may prefer longer, four-week iterations.
  - Naturally, one of the goals of a successful agile team is to reduce (or nearly eliminate) the overhead associated with each iteration.
  - But especially during a team's early iterations, this cost can be significant and will influence the decision about the best iteration length.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

How Soon a Feeling of Urgency Is Established

- “As long as the end date of a project is far in the future, we don’t feel any pressure and work leisurely. When the pressure of the finish date becomes tangible, we start working harder.” - Niels Malotaux (2004).
- Even with four-week iterations , it is sufficiently far away that many teams will feel tangibly less stress during their first week than during the fourth and final week of an iteration.
- The point is not to put the team under more pressure but distribute it more evenly across a suitably long iteration.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Stick with It to Achieve a Steady Rhythm

- Whatever duration you choose, you are better off choosing a duration and sticking with it rather than changing it frequently.
- Teams fall into a natural rhythm when using an unchanging iteration duration.
- A regular iteration rhythm acts like a heartbeat for the project.
- “Rhythem is a significant factor that helps achieve a sustained pace”

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Making a Decision

- One of the main goals in selecting an iteration length is finding one that encourages everyone to work at a consistent pace throughout the iteration.
- If the duration is too long, there is a natural tendency to relax a bit at the start of the iteration, which leads to panic and longer hours at the end of the iteration. Strive to find an interation duration that smooths out these variations.
- Two-week iterations to be ideal.
- Mike Cohen suggests:
  - To follow a macro-cycle of six two-week iterations followed by a one-week iteration. “6 × 2 + 1.”
  - During the one-week iteration, however, the team chooses its own work.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Estimating Velocity

- **It is better to be roughly right than precisely wrong.”—**John Maynard Keynes.
- One of the challenges of planning a release is estimating the velocity of the team. You have the following three options:
  - Use historical values.
  - Run an iteration
  - Make a forecast.
- You should consider expressing the estimate as a range.
  - You could create a range by simply adding and subtracting a few points to the average or by looking at the team’s best and worst iterations over the past two or three months.
  - Example: If your team velocity is 20 story points - You have a very limited chance of being correct in future. Instead give a range 15-24 story points.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

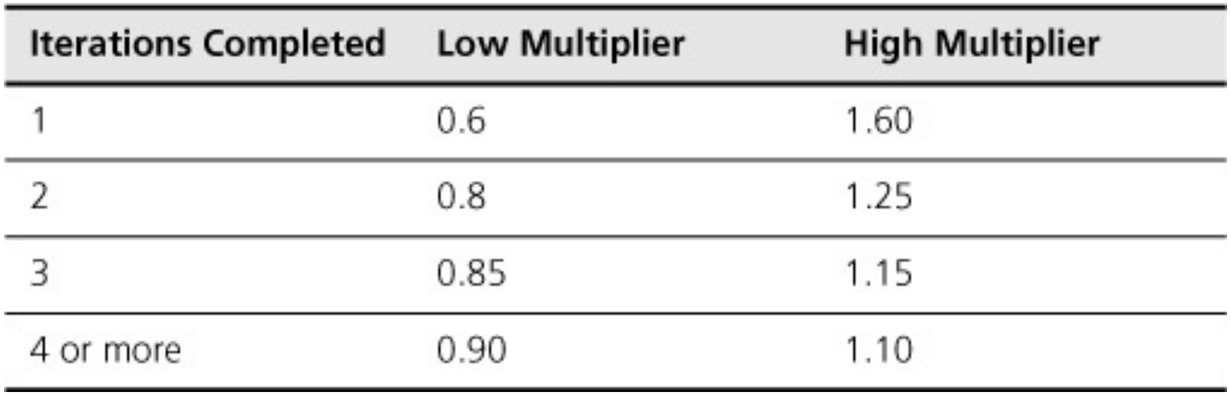
Using Historical values

- Use historical values only when very little has changed between the old project and team and the new project and team.
- Before using them, ask yourself questions like these:
  - Is the technology the same?
  - Is the domain the same?
  - Is the team the same?
  - Is the product owner the same?
  - Are the tools the same?
  - Is the working environment the same?• Were the estimates made by the same people?
  - The answer to each question is often yes when the team is moving onto a new release of a product they just worked on. In that case, using the team's historical values is entirely appropriate. Even though velocity in a situation like this is relatively stable, you should still consider expressing it as a range.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Run an Iteration

- An ideal way to forecast velocity is to run an iteration (or two or three) and then estimate velocity from the observed velocity during the one to three iterations.
- Because the best way to predict velocity is to observe velocity, this should always be your default approach.
- Multipliers for Estimating Velocity Based on Number of Iterations Completed



Iterations Completed	Low Multiplier	High Multiplier
1	0.6	1.60
2	0.8	1.25
3	0.85	1.15
4 or more	0.90	1.10

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Forecasting Velocity

1. Estimate the number of hours that each person will be available to work on the project each day.
2. Determine the total number of hours that will be spent on the project during the iteration.
3. Arbitrarily and somewhat randomly select stories, and expand them into their constituent tasks.
   - Repeat until you have identified enough tasks to fill the number of hours in the iteration.4. Convert the velocity determined in the preceding step into a range.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Example

Number of Team members	Available hours per day/team member	Total Available hrs	Team Capacity for 10 days iteration
4	6 hrs.	4*6= 24	10*24=240

Product Backlog	
As a Cab Driver, I Want to	3
As a Cab Driver, I Want to	5
As a Cab Driver, I Want to	5
As a Cab Driver, I Want to	2
As a Cab Driver, I Want to	5

60hrs

60hrs

22hrs

60hrs

Sprint Backlog	
Code the UI	10
Write Tests	8
Write DB Stored Procedure	12
Automate Tests	10

40hrs

Velocity = 3+5+5+2+5=20 Story Points

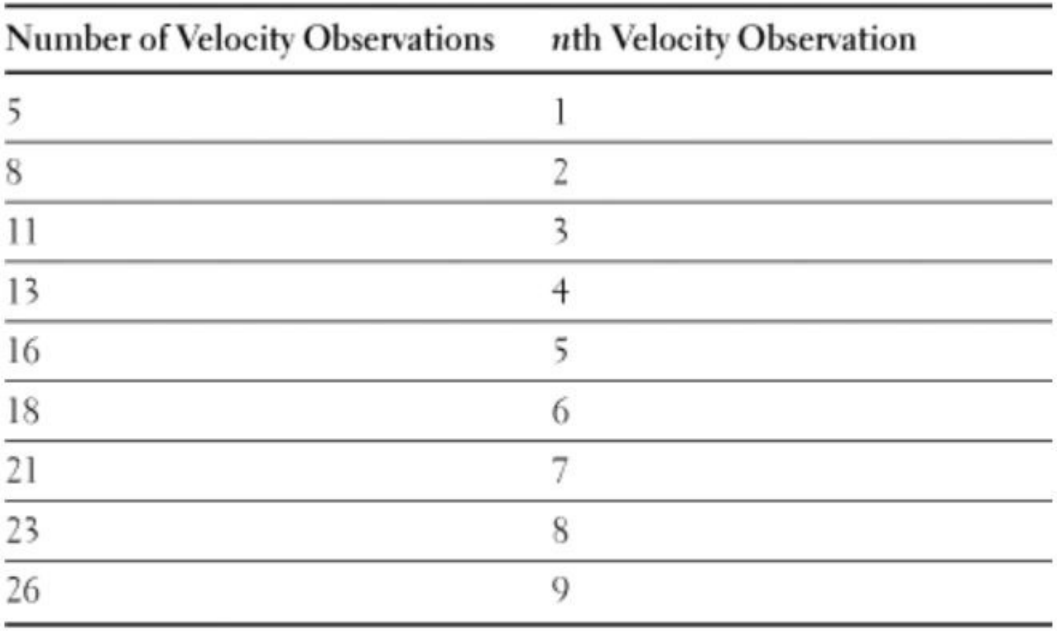
Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

High-confidence forecast- Example

- Suppose we want to create high confidence forecast(90%) for the next release.
- As soon as the team has run five or more sprints, we can create a high-confidence forecast
- Suppose, Velocity of completed 8 sprints:20, 25, 28, 26, 16, 20, 26, 26.
- Sorted list:16, 20, 20, 25, 26, 26, 26, 28

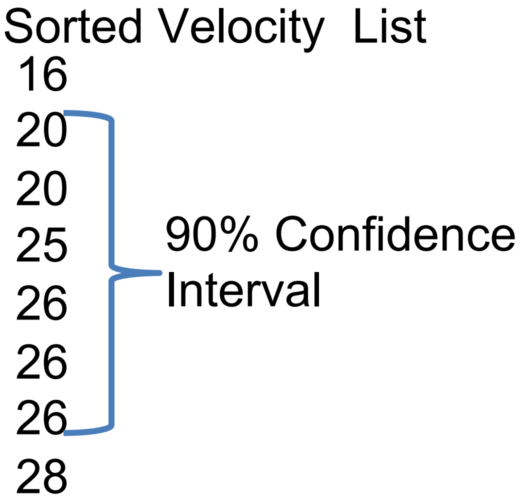
Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Use the nth Lowest and the nth Highest Observation of a Sorted List of Velocities to Find a 90% Confidence Interval



Number of Velocity Observations	*n*th Velocity Observation
5	1
8	2
11	3
13	4
16	5
18	6
21	7
23	8
26	9

- Velocity of completed **8 sprints**
- :20, 25, 28, 26, 16, 20, 26, 26



Sorted Velocity List
16
20
20
25
26
26
26
28

90% Confidence Interval

20 ➔ Lower confidence, certainly we will do
23 ➔ Mean Velocity – We will get here
26 ➔ Upper confidence, Most we could expect

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

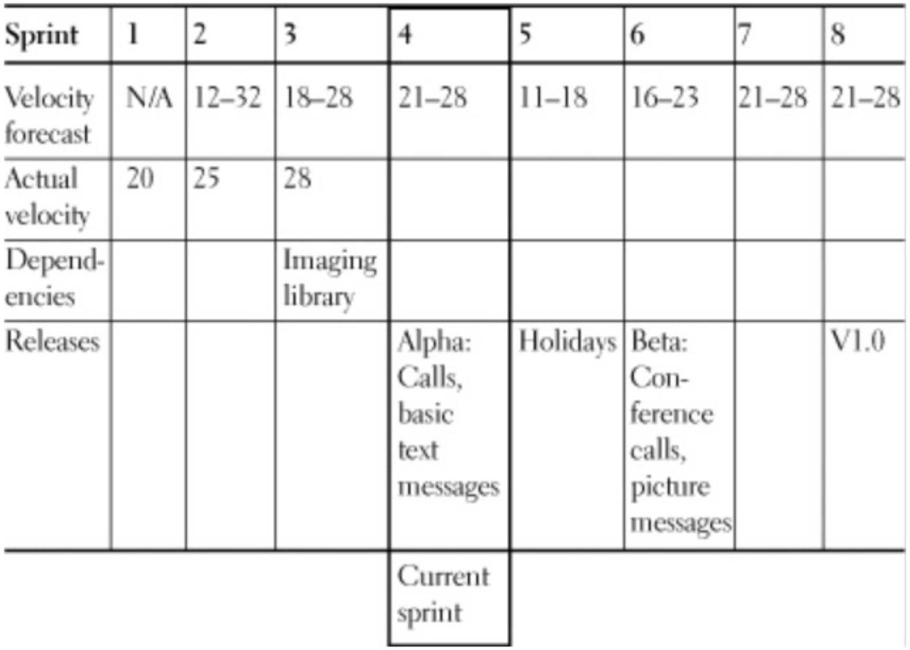
Source: Succeeding with Agile SW development by Mike Cohen

Creation a Release Plan

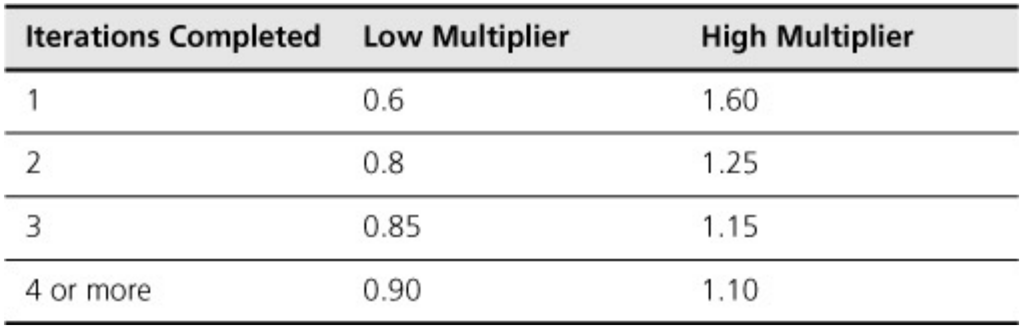
- Total story points of Release backlog **divided by** Mean velocity or Velocity range.
- This will give us a provisional number of sprints required for the release
- Example:
  - Total Story points = 200; Mean velocity = 20;Number Sprints required = 10
- We then map the identified number of sprints onto the calendar and consider the factors that are likely to influence the velocity and that are not accounted for in the velocity forecast.
  - These can include holidays, vacations, training and development, sickness statistics, and planned changes to the project organization, such as modifying the team composition. We adjust the forecasted velocity of each sprint accordingly.

Ref: Agile Estimating and Planning by Mike Cohn Published by Addison-Wesley Professional, 20

Sample Release Plan



Sprint	1	2	3	4	5	6	7	8
Velocity forecast	N/A	12–32	18–28	21–28	11–18	16–23	21–28	21–28
Actual velocity	20	25	28					
Dependencies			Imaging library					
Releases				Alpha: Calls, basic text messages	Holidays	Beta: Conference calls, picture messages		V1.0
				Current sprint				

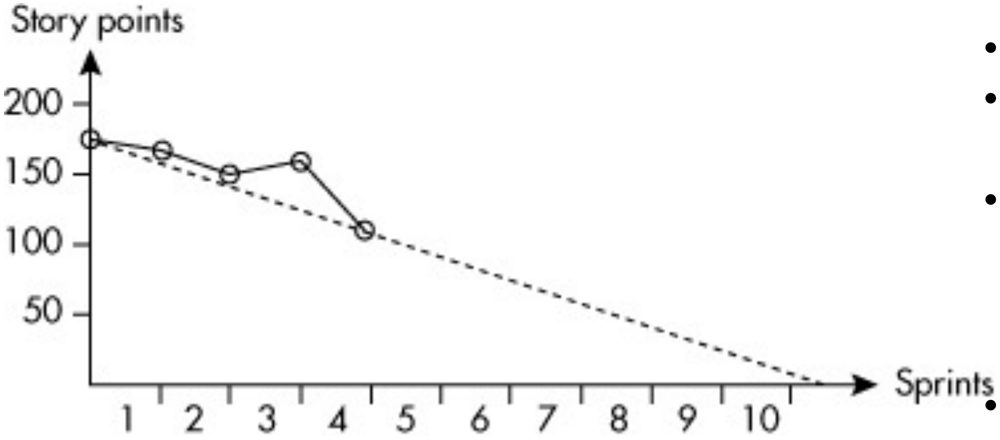


Iterations Completed	Low Multiplier	High Multiplier
1	0.6	1.60
2	0.8	1.25
3	0.85	1.15
4 or more	0.90	1.10

1. Actual velocity for the first three sprints of 20, 25, and 28.
2. The average (mean) velocity per sprint, then, is 24 points.
3. The Scrum team has forecasted a velocity of 21 to 28 points for the fourth, seventh, and eighth sprints using the multipliers Table(below).
4. The release plan also anticipates a velocity drop in sprints five and six, when several team members will take time off and then return to work.

Source: Agile Estimation and Planning by Mike Cohen

Tracking the Progress of the Release – Release Burndown chart



- The solid line is the actual burndown- Indicate the progress
- Slow start. Might be - impediments and risks materializing, team-building dynamics, or technology issues.
- Third sprint, the remaining effort even increased. - caused by the team reestimating backlog items or discovering new requirements
- The fourth sprint saw a steep burndown; the project progressed fast.

- X-Axis - Number of sprints as the unit
- Y- Axis - Number of story points estimated
- The first data point is the estimated effort of the entire product backlog before any development has taken place.
- To arrive at our next data point, we determine the remaining effort in the product backlog at the end of the first sprint.
- Then we draw a line through the two points. This line is called the burndown(…. Line)
- It shows the rate at which the effort in the product backlog is consumed.
- If we extend the burndown line to the x-axis, we can forecast when the project is likely to finish—assuming effort and velocity stay stable.

Product Visioning - Level 1

- A product vision describes the **future state of a product** (Big Picture) that a company or team desires to achieve. You can also define that future state as: **a goal**.
- Aligns:
  - Product strategy, product development roadmap, backlog & planning, execution & product launch
  - There is/can be a difference between a product and company vision.
- What information does a product vision contain?
  - Focused on Customers (B2C or B2B)- How it will benefit the company and the customer.(What?)
  - It's looking into the future and outlining a clear state of the product/goal that the company and team(s) want to achieve. This goal should be underlined with the motivation behind it (Why?, not How?)
  - The art of defining a great product vision that people want to follow is to make it catchy.

Source:https://www.christianstrunk.com/blog/product-vision

A. How to define a product vision?

1. Defining key product information.
   - Have some valid data in the product discovery process to find answers to open questions.
   - Gaining a clear picture of your customer, your market, the problems you want to solve, and your business goals
   - According to Roman Pichler's product vision board, it's important to answer 4 key questions:
   - What's the target group?, What are the customer needs?, What is and will be the product and it's USP(s)?,What are the business goals?
2. Phrasing the product vision in one inspiring sentence.
   - Examples: Google's companies vision statement is: "*to provide access to the world's information in one click.*" because that's Google's core business.
   - Card reader Makers: "*We believe in a world where small businesses can offer a super fast and safe payment experience to their customers, for minimal costs with no administrative efforts.*"

3. Why is having a product vision important? – Gives direction to Teams

- Who owns? What is the Process to create product vision?

B. How to define a product vision?- Classical format.

- **Create an elevator statement** or a Product vision box/Product Vision Board . (Non technical)
- A format popularized by Geoffrey Moore's classic *Crossing the Chasm*

For **(target customer)** who **(statement of need or opportunity)**, the **(product name)** is a **(product category)** that **(key benefit, reason to buy)**.

Unlike **(primary competitive alternative)**, our product **(statement of primary differentiation)**.

- Here's an example of a product vision statement for Microsoft Surface:
  - For **the business user** who **needs to be productive in the office and on the go**, the **Surface** is a **convertible tablet** that **is easy to carry and gives you full computing productivity no matter where you are**.
    Unlike **laptops**, **Surface serves your on-the-go needs without having to carry an extra device**.
- Any further planning (Design) at this stage may divert our attention from future vision of the product.

Source:280 Group LLC

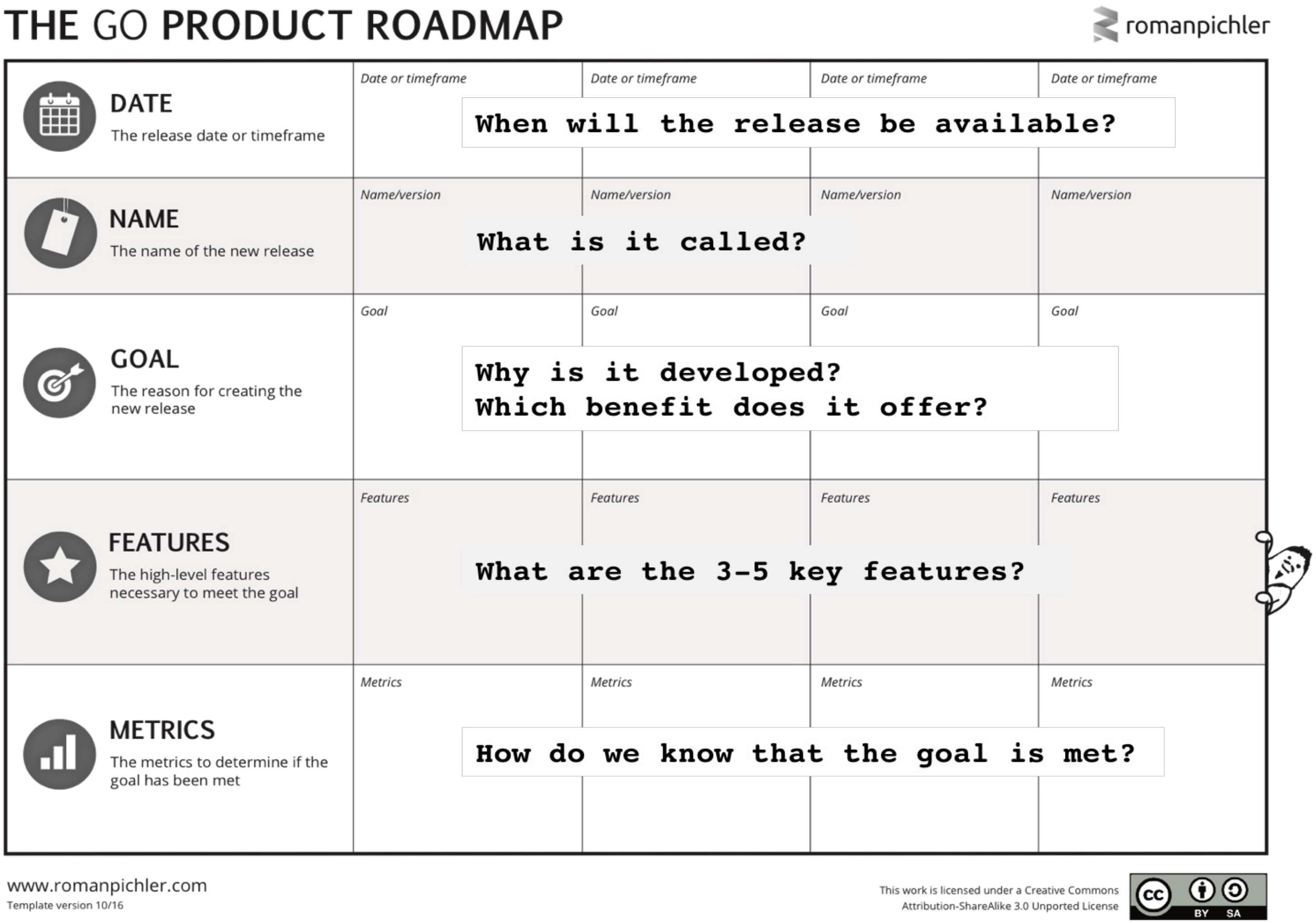
Product Roadmap – Level 2 Planning

- A product vision is a high-level aspirational projection of the future state of a product.
  - It must be impactful to generate sufficient interest among the innovators, early adopters, and early-stage investors.
- A product roadmap is essentially a timeline of feature rollout plans.- (Review the roadmap regularly)
  - It helps product managers prioritize R&D dollars to maximize chances of realizing the product's promised or anticipated ROI.
  - It allows the product team to focus on more value-creating features "here and now" versus hundreds of features that might have limited relative potential.
  - It helps customers know that their favorite features are planned somewhere down the road, and, if they so desire, the product team can expedite them.
  - it also allows customer feedback of what features are perceived as critical and what could be deferred to another time.
  - Helps the delivery team to see as whole, learn business priorities, Provide technical and estimates inputs to Product roadmap.

Type of Product Roadmaps

- Goal/Objectives driven roadmap
- Feature Driven
- Date/Time Driven

Goal Oriented Roadmap



THE GO PRODUCT ROADMAP

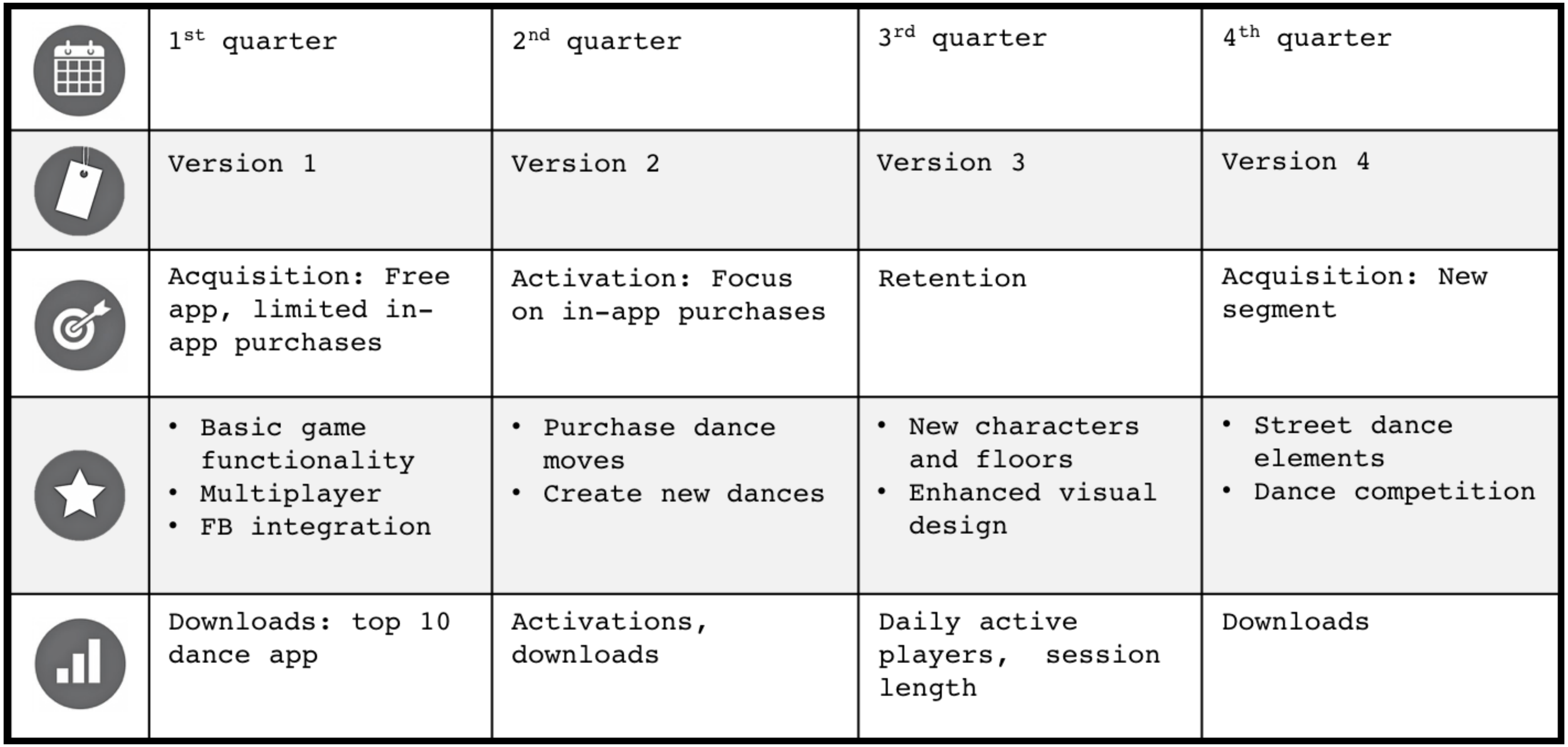
romanpichler

DATE The release date or timeframe	*Date or timeframe* When will the release be available?	*Date or timeframe*	*Date or timeframe*	*Date or timeframe*
NAME The name of the new release	*Name/version* What is it called?	*Name/version*	*Name/version*	*Name/version*
GOAL The reason for creating the new release	*Goal* Why is it developed? Which benefit does it offer?	*Goal*	*Goal*	*Goal*
FEATURES The high-level features necessary to meet the goal	*Features* What are the 3-5 key features?	*Features*	*Features*	*Features*
METRICS The metrics to determine if the goal has been met	*Metrics* How do we know that the goal is met?	*Metrics*	*Metrics*	*Metrics*

www.romanpichler.com
Template version 10/16

Goal Oriented Roadmap – An Example

Develop a new dance game for girls aged eight to 12 years. The app should be fun and educational allowing the players to modify the characters, change the music, dance with remote players, and choreograph new dances.

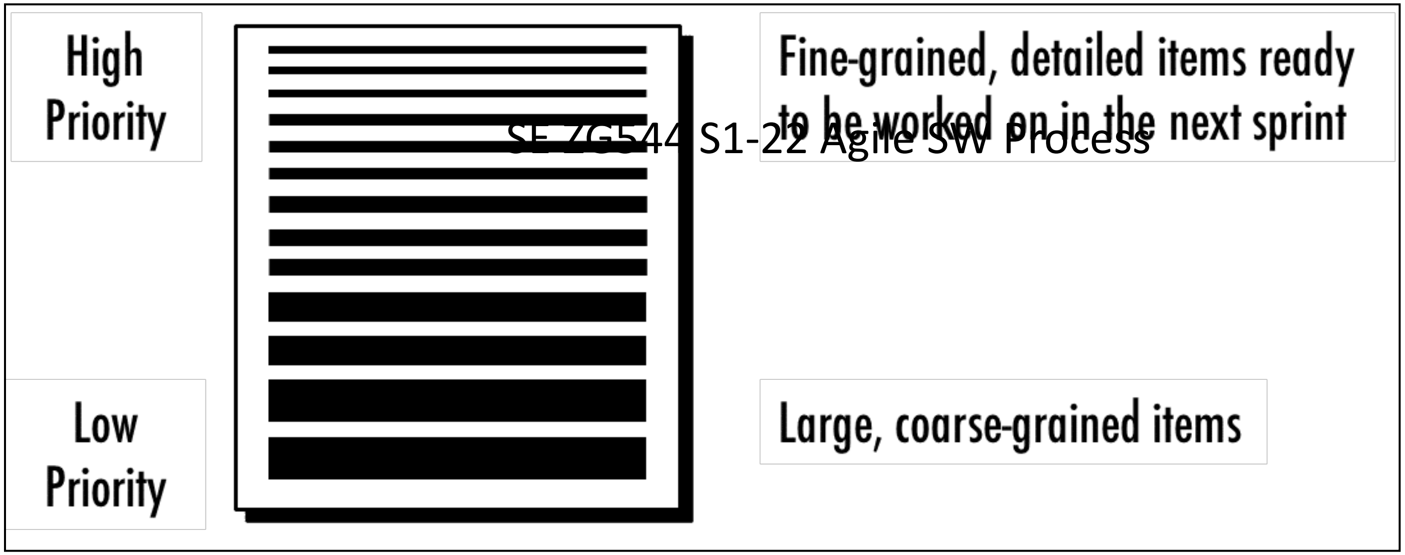


📅	1st quarter	2nd quarter	3rd quarter	4th quarter
🏷	Version 1	Version 2	Version 3	Version 4
🎯	Acquisition: Free app, limited in-app purchases	Activation: Focus on in-app purchases	Retention	Acquisition: New segment
★	• Basic game functionality • Multiplayer • FB integration	• Purchase dance moves • Create new dances	• New characters and floors • Enhanced visual design	• Street dance elements • Dance competition
📊	Downloads: top 10 dance app	Activations, downloads	Daily active players, session length	Downloads

Source:https://www.romanpichler.com/blog/goal-oriented-agile-product-roadmap/

Product Backlog - Level 3

- A product backlog is a prioritized list of work* for the development team that is derived from the roadmap and its requirements.
- The most important items are shown at the top of the product backlog so the team knows what to deliver first.


High Priority
Low Priority
Fine-grained, detailed items ready to be worked on in the next sprint
Large, coarse-grained items

The product backlog is the responsibility of the product owner who cares about content, availability, and priority of the backlog.

* List of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.

Source:https://www.romanpichler.com/blog/goal-oriented-agile-product-roadmap/

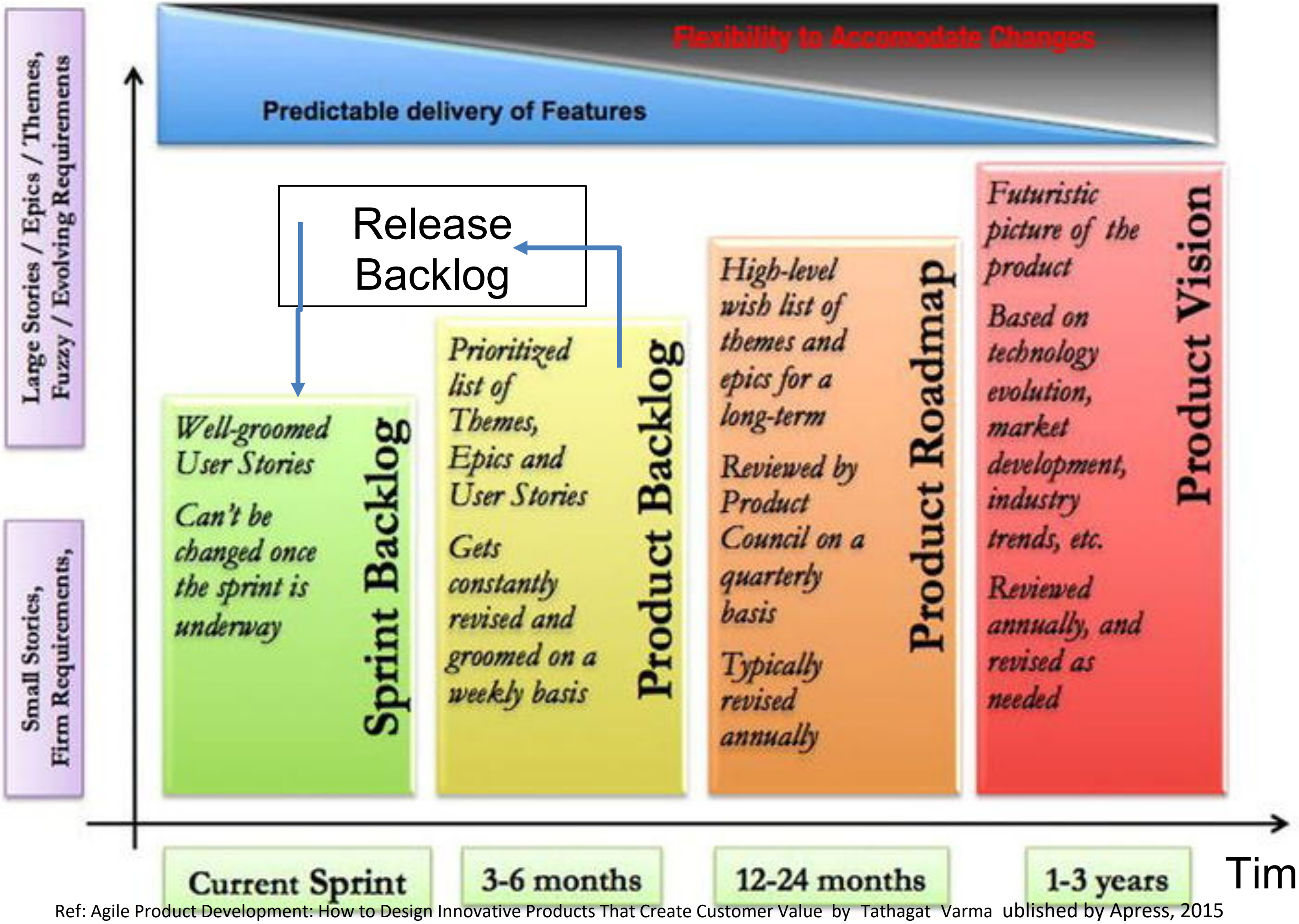
Characteristics of a Product Backlog

- There is an abbreviation that combines similar characteristics of good product backlogs. This is DEEP:
- **D**etailed appropriately
  - higher-priority items are described in more detail than lower-priority ones
- **E**mergent
  - It evolves and its contents change frequently. New items emerge based on customer and user feedback and are added to the backlog. Existing items are modified, reprioritized, refined, or removed on a regular basis.
- **E**stimated
  - The product backlog items–certainly the ones participating in the next major release–should be estimated. The estimates are coarse-grained and often expressed in story points or ideal days.
- **P**rioritized
  - All items in the product backlog are prioritized (or ordered)

Source: https://www.romanpichler.com/blog/make-the-product-backlog-deep/

Product runways represent a healthy trade-off between flexibility and predictability

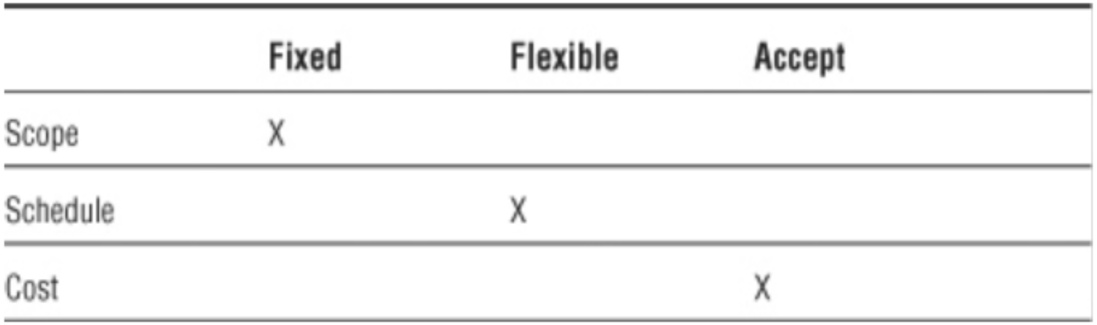
Flexibility to accommodate change


Flexibility to Accomodate Changes
Predictable delivery of Features
Large Stories / Epics / Themes, Fuzzy / Evolving Requirements
Small Stories, Firm Requirements,
Release Backlog
Sprint Backlog
Well-groomed User Stories
Can't be changed once the sprint is underway
Product Backlog
Prioritized list of Themes, Epics and User Stories
Gets constantly revised and groomed on a weekly basis
Product Roadmap
High-level wish list of themes and epics for a long-term
Reviewed by Product Council on a quarterly basis
Typically revised annually
Product Vision
Futuristic picture of the product
Based on technology evolution, market development, industry trends, etc.
Reviewed annually, and revised as needed
Current Sprint
3-6 months
12-24 months
1-3 years

Timeline vary

Ref: Agile Product Development: How to Design Innovative Products That Create Customer Value by Tathagat Varma ublished by Apress, 2015

Project Trade-off Matrix



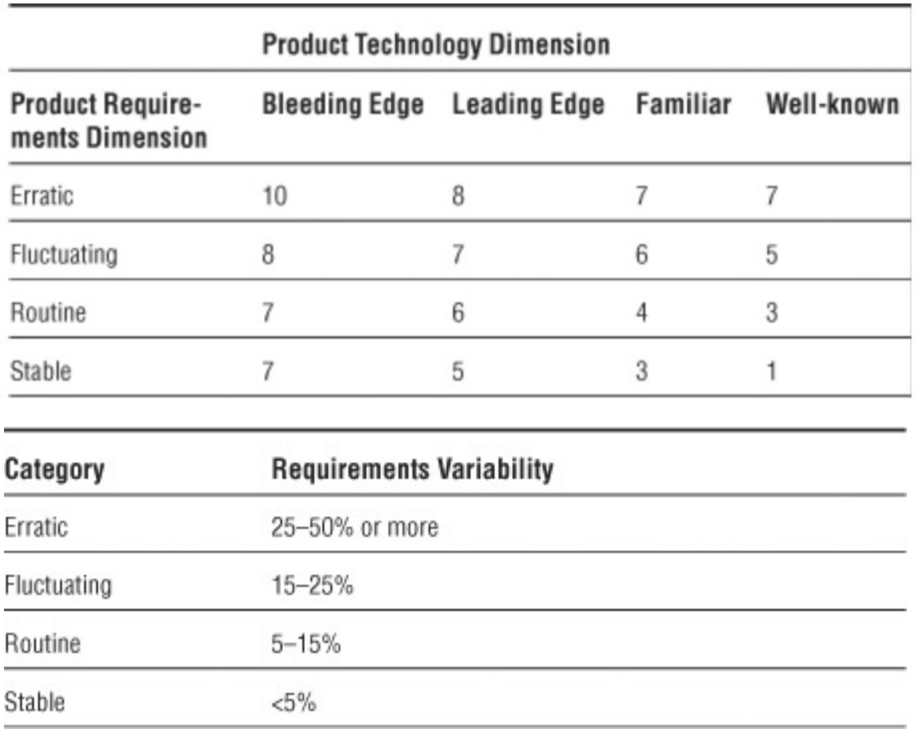
	Fixed	Flexible	Accept
Scope	X		
Schedule		X	
Cost			X

- The tradeoff matrix helps the development team, the product team, and the executive stakeholders manage change during a project.
- The trade-off matrix informs all participants that changes have consequences and acts as a basis for decision making.
- The trade-off matrix indicates relative importance of the three constraints (scope, schedule, cost) identified on the agile triangle (value, quality, constraints).
- The importance goes from Fixed, to Flexible, to Accept, the tolerance for variation increases.

Ref: Agile Project Management: Creating Innovative Products, Second Edition by Jim Highsmith Published by Addison-Wesley Professional, 2009

Exploration Factor

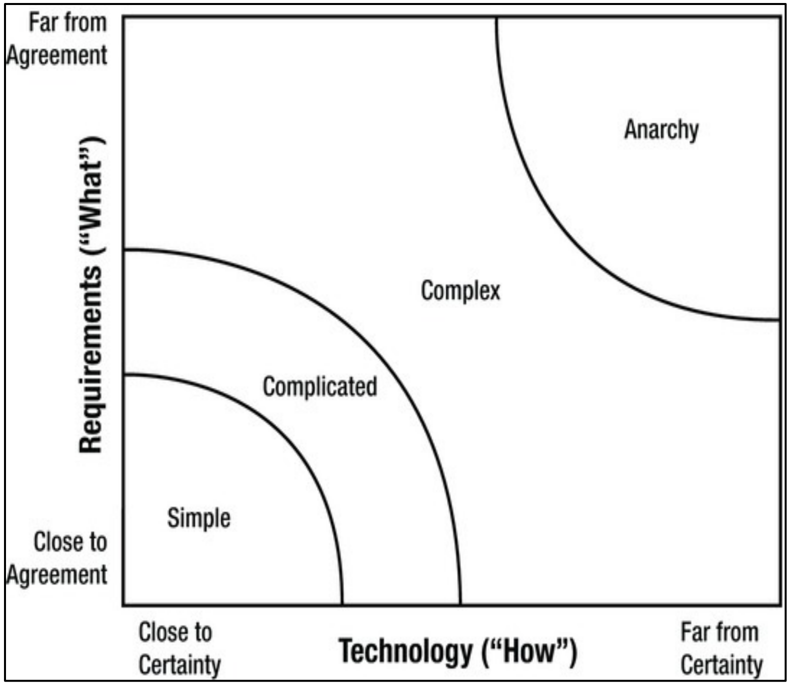
- Articulating an exploration factor helps considerably in managing customer and executive expectations.



	Product Technology Dimension			
Product Requirements Dimension	Bleeding Edge	Leading Edge	Familiar	Well-known
Erratic	10	8	7	7
Fluctuating	8	7	6	5
Routine	7	6	4	3
Stable	7	5	3	1

Category	Requirements Variability
Erratic	25–50% or more
Fluctuating	15–25%
Routine	5–15%
Stable	<5%

Stacey Matrix



Ref: Agile Project Management: Creating Innovative Products, Second Edition by Jim Highsmith Published by Addison-Wesley Professional, 2009

End