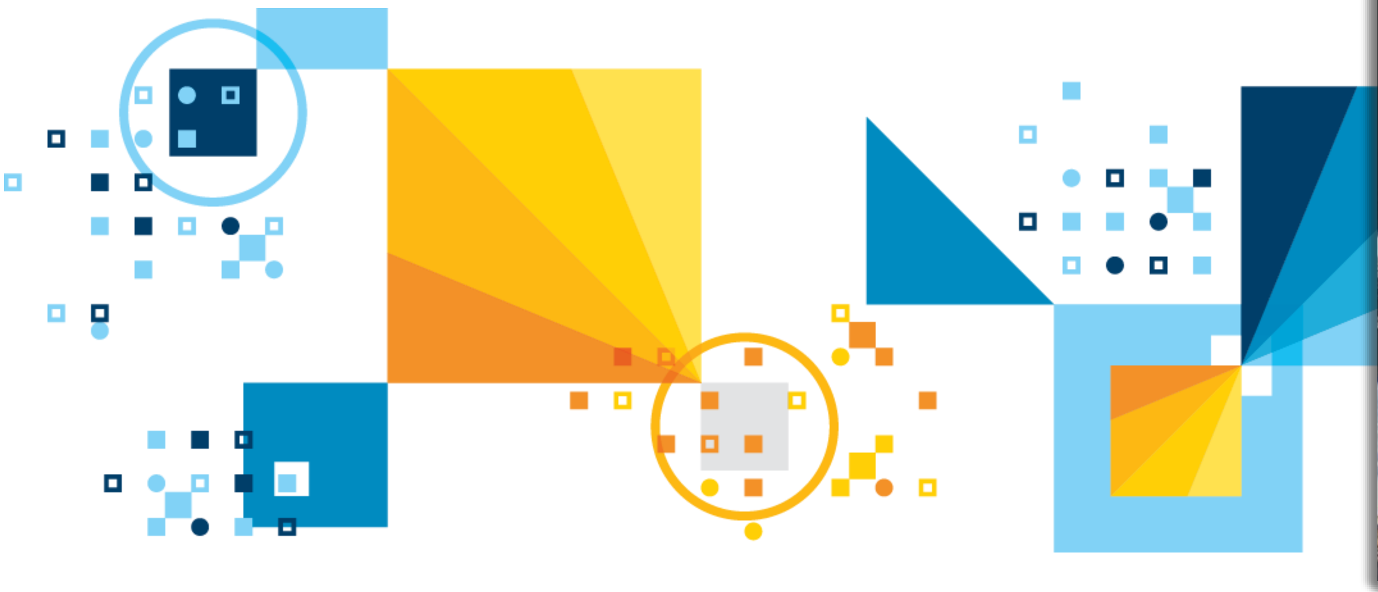
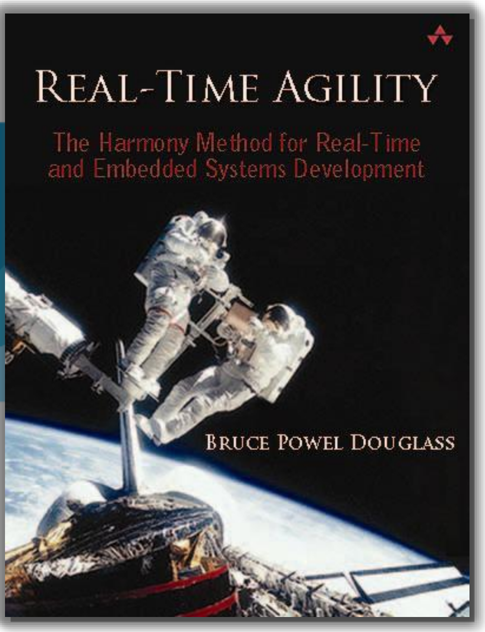
Agile Planning for Embedded Software Development

Bruce Powel Douglass, Ph.D.
Chief Evangelist, IBM IoT
Bruce.Douglass@us.ibm.com
Twitter: @IronmanBruce
www.bruce-douglass.com





REAL-TIME AGILITY

The Harmony Method for Real-Time and Embedded Systems Development

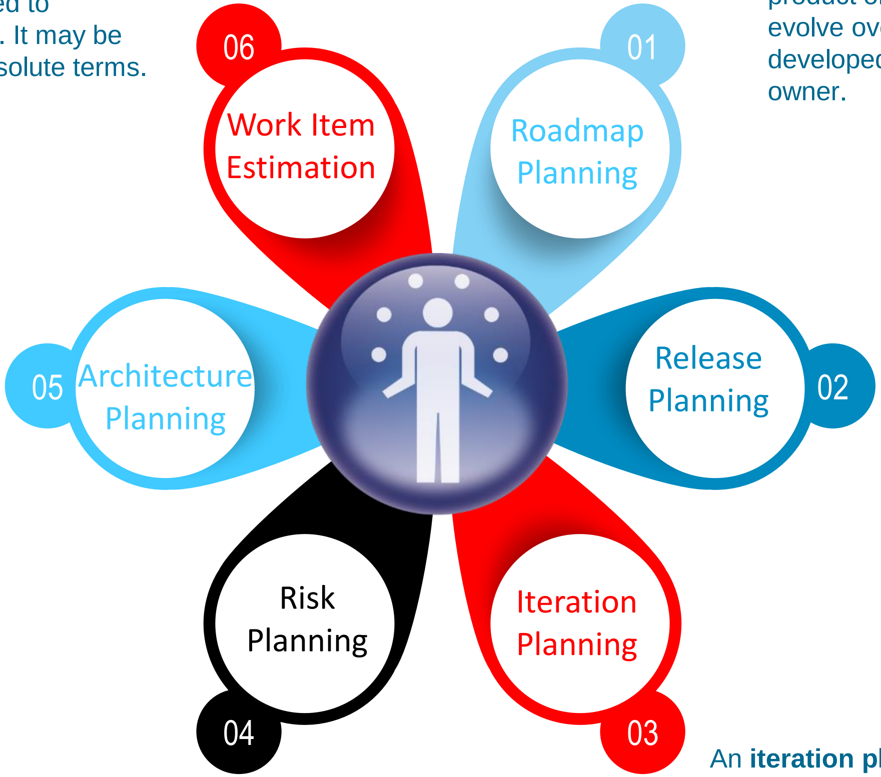
BRUCE POWEL DOUGLASS

© 2019 Bruce Powel Douglass, Ph.D.

Kinds of Agile Planning

Work item estimation appraises the work effort required to complete a work item. It may be done in relative or absolute terms.

A **product roadmap** is a plan of action for how a product or solution will evolve over time developed by the product owner.



An **architecture plan** organizes the overall structuring and optimization decisions for the software

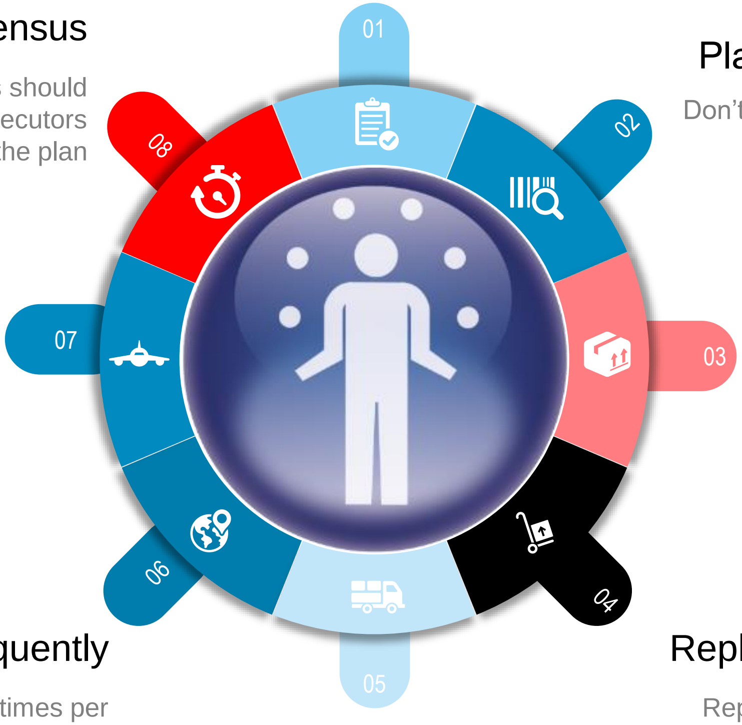
A **release plan** maps features and user stories to a set of product releases

An **iteration plan** details the work to be done in the upcoming iteration

An **risk plan** identifies risks to successful project completion, characterizes them, and plans for their resolution in a series of spikes mapped into the iterations

© 2019 Bruce Powel Douglass, Ph.D.

Key Agile Planning Concepts



Be open to change

A key principle of Agile is to “embrace change”

Plan “just enough”

Don’t plan beyond your level of information

Plan “just in time”

Plan just before you need it to be sure your information is an accurate and complete as possible

Replan frequently

Replan as evidence is collected so that plans are as accurate and complete as possible

Decide how to measure success

Measure success against goals of the plan

Measure frequently

Measure multiple times per task scope.

Time box or feature box

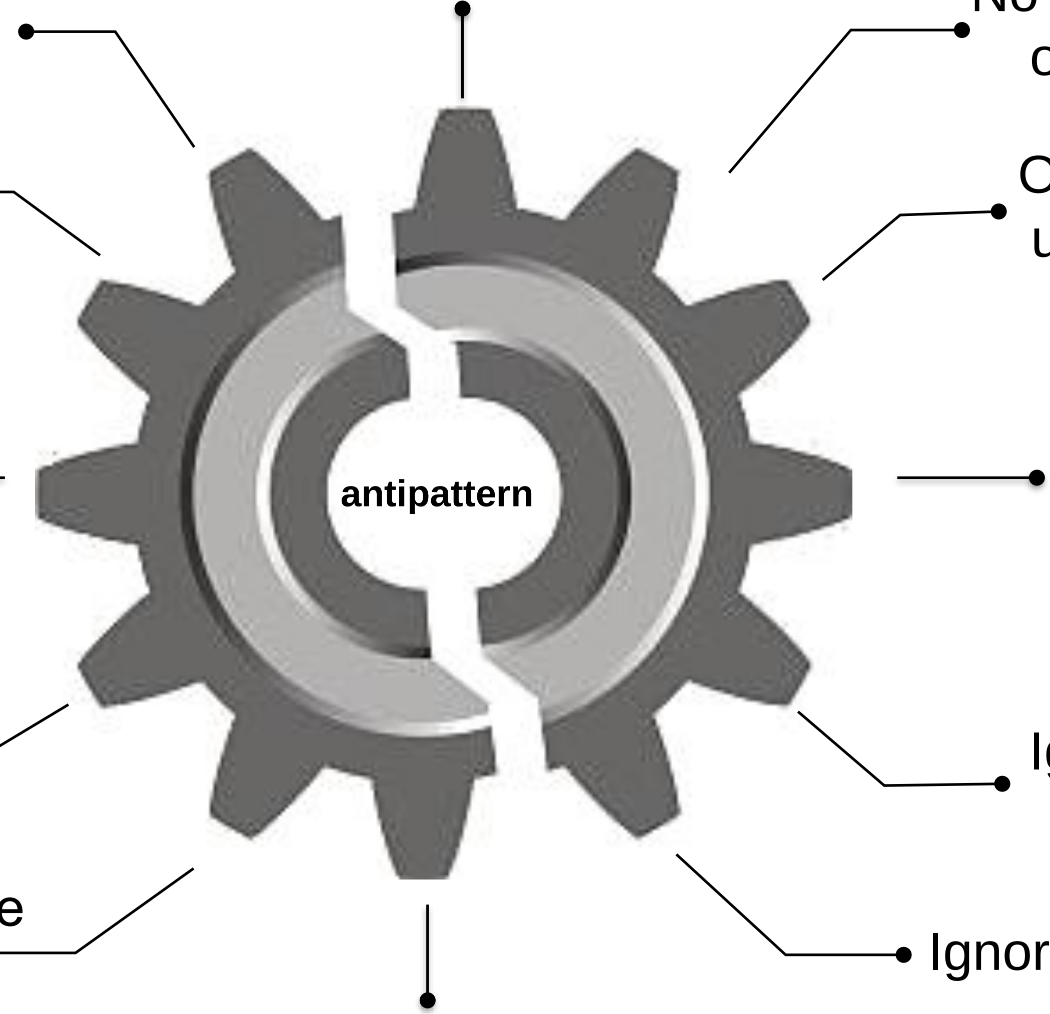
This ensures both progress and allows collaboration.

Achieve consensus

All involved parties should agree, including the executors of the plan

© 2019 Bruce Powel Douglass, Ph.D.

Planning anti-patterns



antipattern

Ignore previous outcomes

No Definition of Done

Over promise, under deliver

Add no slack

Ignore risk factors

Ignore technical debt

No consensus

Don't measure velocity

Don't measure success

Under plan

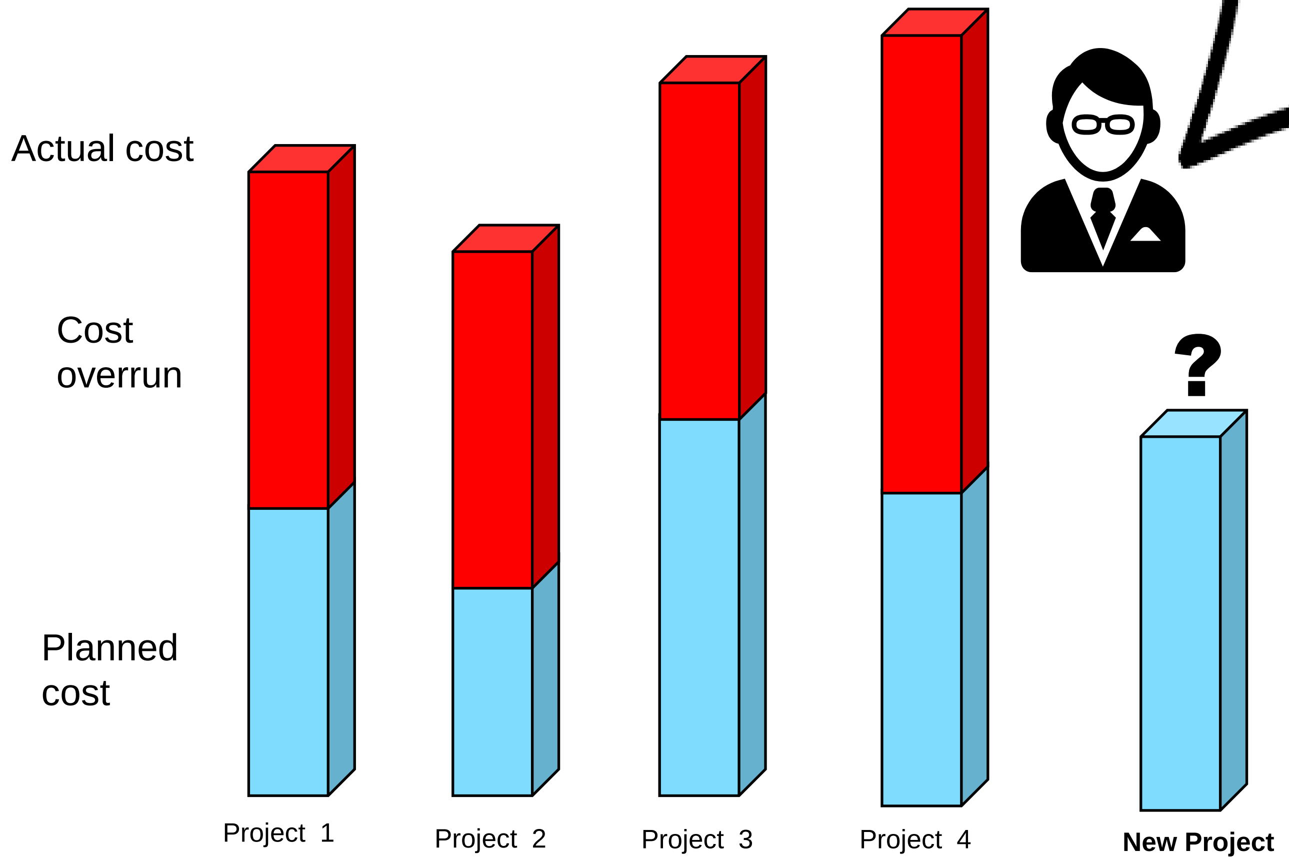
Over plan

Plan once

© 2019 Bruce Powel Douglass, Ph.D.

The 3rd most common mistake in planning

We ignore previous project outcomes

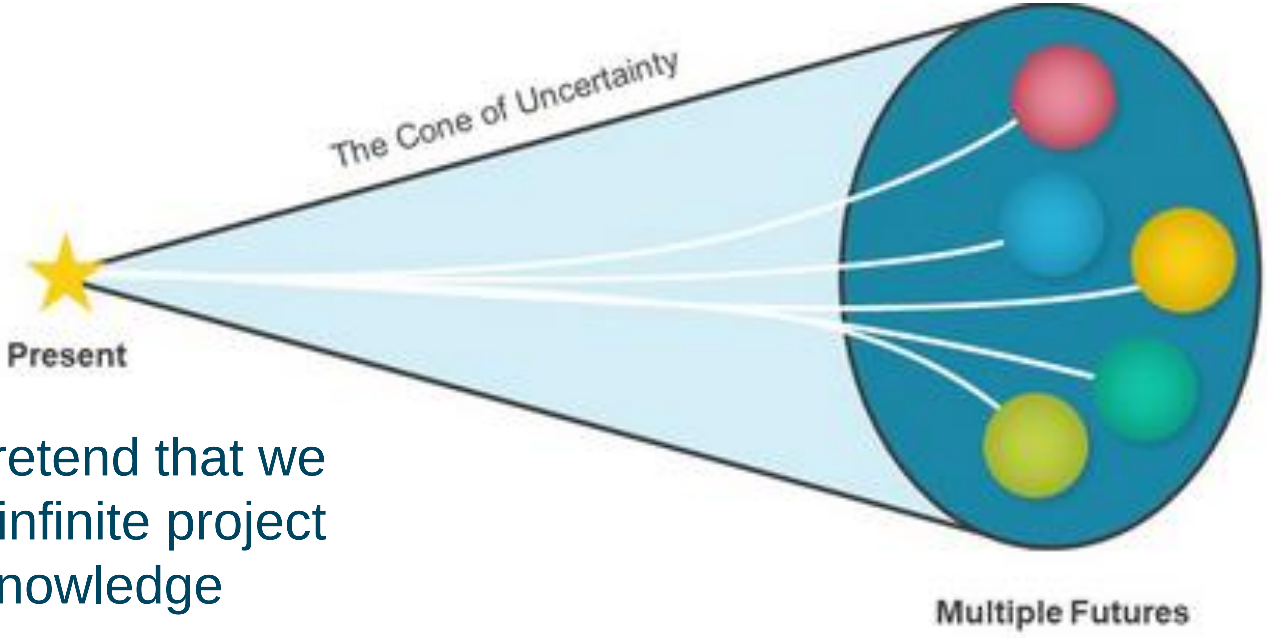


The new project will, of course, come in as planned…

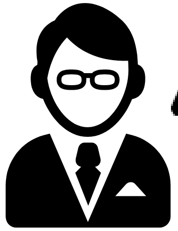
For the first time in our history

© 2019 Bruce Powel Douglass, Ph.D.

The 2nd most common mistake in planning



We pretend that we have infinite project knowledge



The project will complete on May 27, 2021 at 4:36pm … and 13 seconds

But really, we’re estimating things we really don’t know

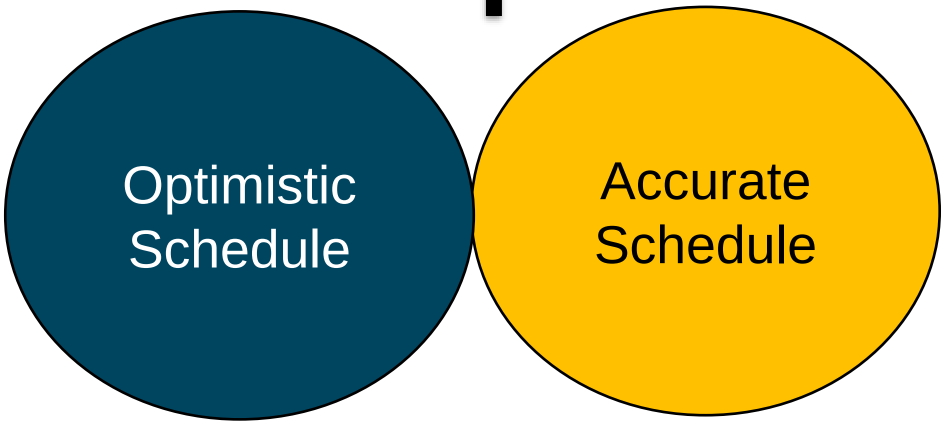
© 2019 Bruce Powel Douglass, Ph.D.

The most common mistake in planning

A schedule may be use to motivate workers (this is known as an “optimistic schedule”

OR

A schedule may be use to accurately describe project cost, effort and time



Optimistic Schedule

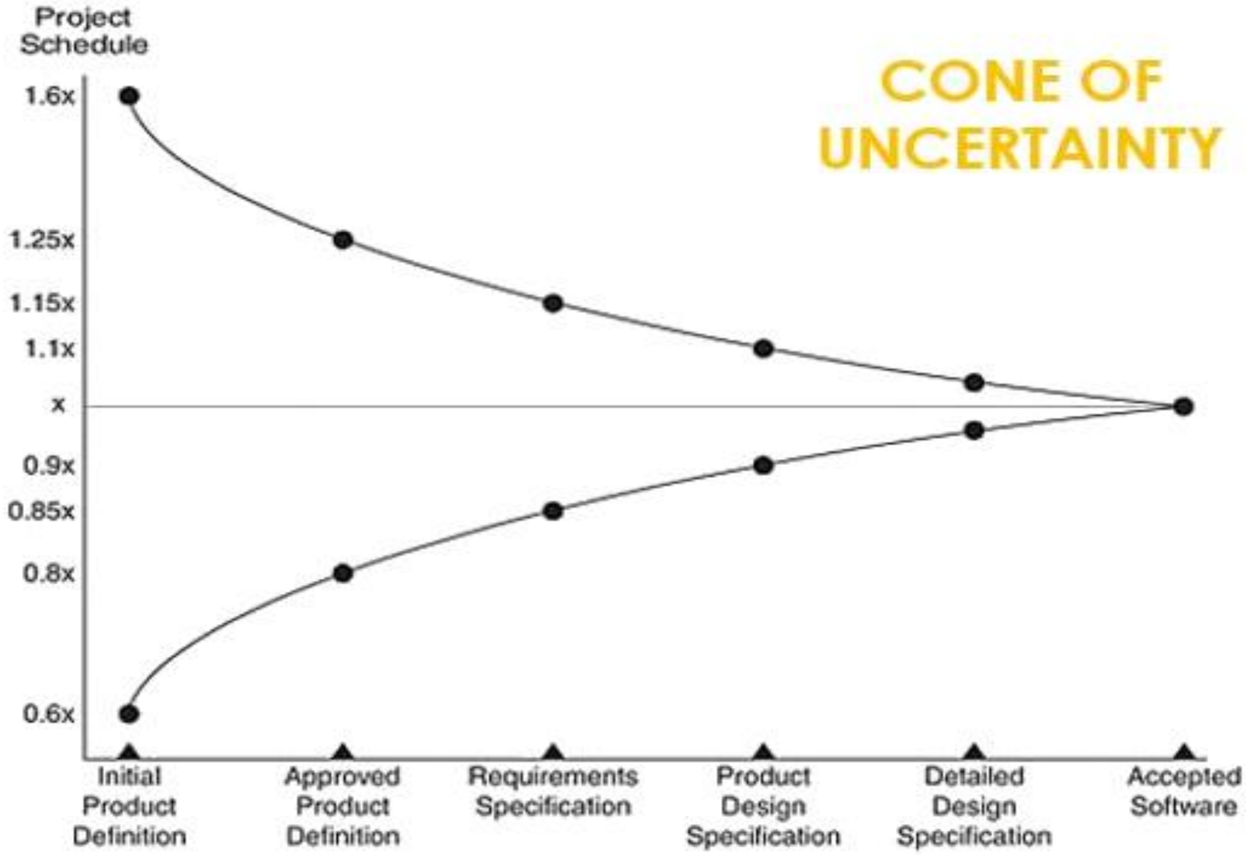
Accurate Schedule

Motivational – Accurate schedule Venn diagram

Optimism is the enemy of realism

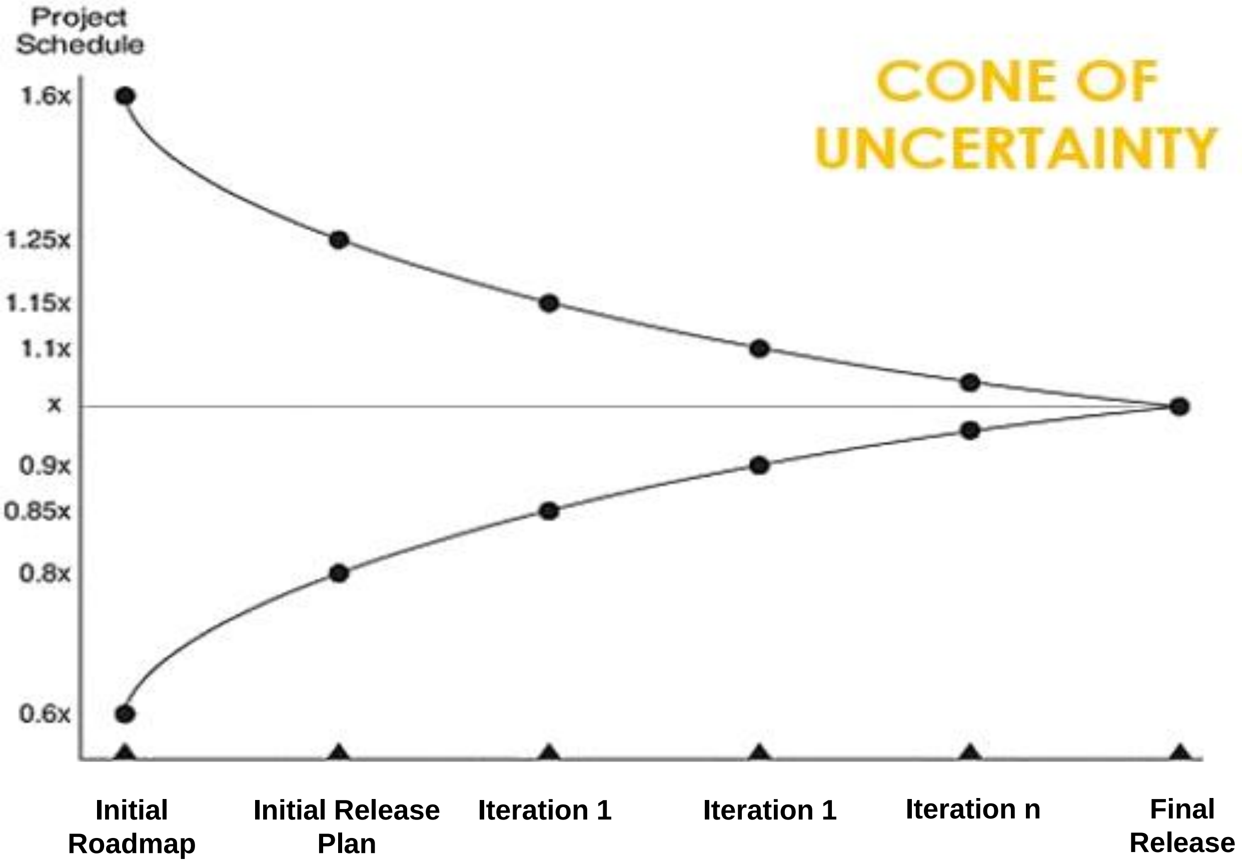
© 2019 Bruce Powel Douglass, Ph.D.

Cone of uncertainty for traditional plans



© 2019 Bruce Powel Douglass, Ph.D.

Cone of uncertainty for agile plans



© 2019 Bruce Powel Douglass, Ph.D.

Ballistic versus dynamic planning

Assumes infinite knowledge

How long, exact effort, start and end times, interruptions, resource constraints, etc

Planning is done once

Plan is constructed early, before analysis, design, specification, funding, tool selections, and team allocation

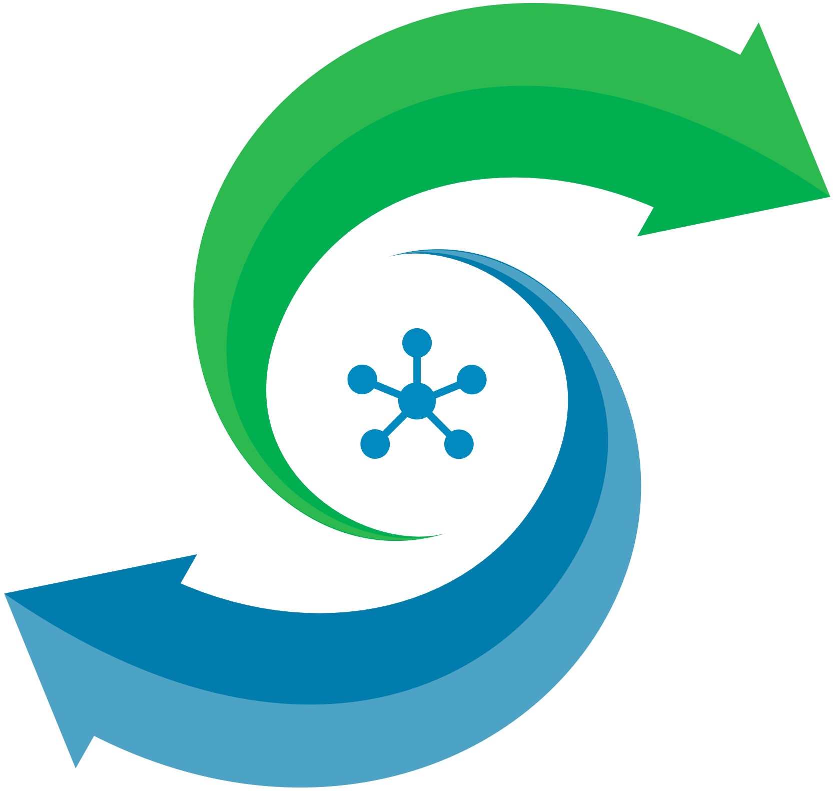
No monitoring is required

Because the plan is assumed to be truth

Result is a *number*

The exact time, effort, and cost is assumed to be represented by the plan

Dynamic Planning



Ballistic Planning

Planning is done with “error bars”

Plans are assumed to incomplete and erroneous and are based on available information

Plan is done many times

The plan increases in accuracy as we gain information and incorporate it into our plan.

Monitoring is necessary

Metrics, such as velocity and burn down rate, are necessary to improve the plan

Result is a *range*

The plan defines a range, the size of which depends on the fidelity and correctness of data

© 2019 Bruce Powel Douglass, Ph.D.

Graphic by PresentationGO.com. Used with permission.

Iteration 0

These work products guide the product development to come

On-board the team and get them ready to begin, including, possible domain or tool training, familiarity with the stakeholders, the tooling, processes, and regulatory needs

Focus on	Work to be done
Product	Create the initial vision, product plan, and release plan
Team	Ready the team with knowledge, skill, tools, and processes
Environment	Install, configure and test tooling; set up work spaces
Architecture	Define the high-level architecture and design goals to guide emergent and incremental delivery of business value

Includes compilers, editors, and requirements, modeling, configuration management and other tools. Also, set up the individual and collaborative work spaces.

Architectural concepts, design constraints, platform characteristics, and optimization criteria will drive a lot of work. While the architecture will change over time, define the starting point.

Iteration 0: Do the work that needs to be done before there is work to do

© 2019 Bruce Powel Douglass, Ph.D.

Agile Roadmap

01

Purpose

The roadmap is a high level view of the series of deliverable systems mapped to capabilities and customer needs.

02

Description

The product owner take into account market trajectories, value propositions, and engineering constraints. Once these factors are reasonably well-understood, they are expressed in a roadmap as initiatives and timelines.

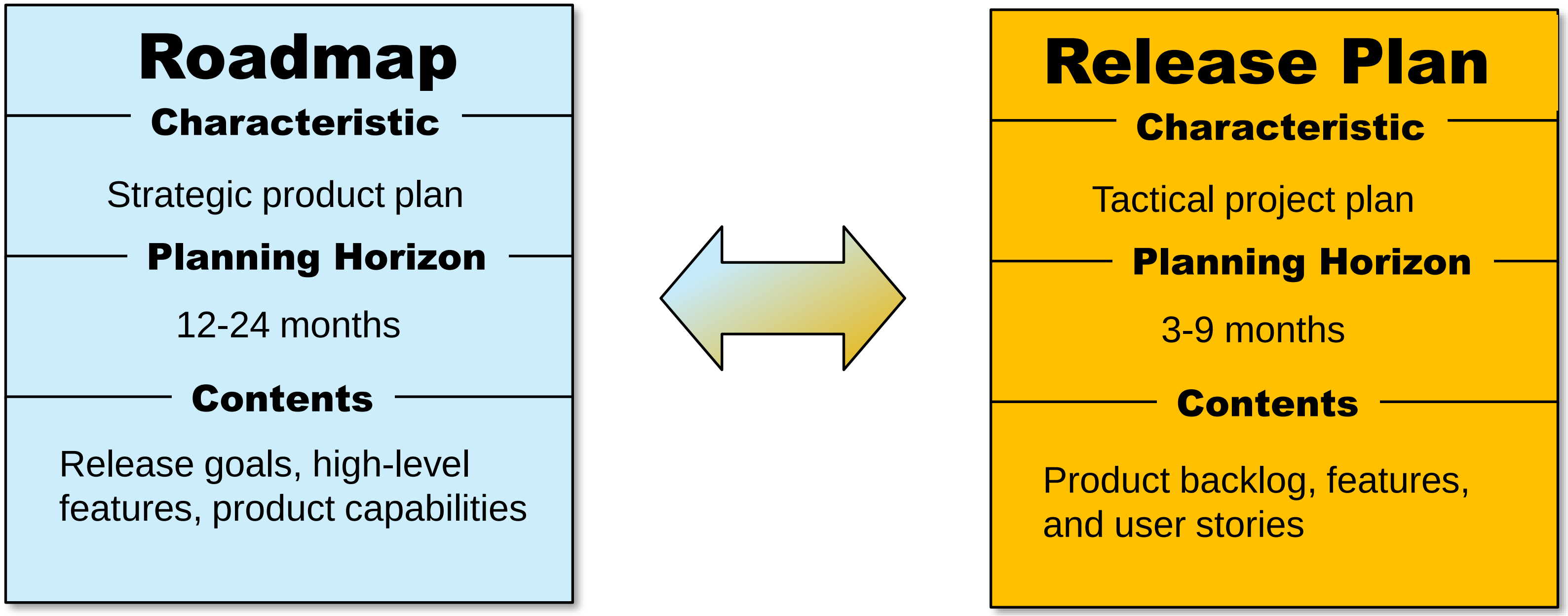
03

Hints

1. Initial roadmap is usually done in a day or less
2. Roadmap and release plans are updated continuously – or at least very frequently.
3. Note that an iteration produces an **increment** which is a potential release, while a **release** is delivered to relevant stakeholders.
4. This is done prior to the start of the actual development work, during a period of time often called **iteration 0**

© 2019 Bruce Powel Douglass, Ph.D.

Roadmap or Release Plan?



Roadmap

Characteristic

Strategic product plan

Planning Horizon

12-24 months

Contents

Release goals, high-level features, product capabilities

Release Plan

Characteristic

Tactical project plan

Planning Horizon

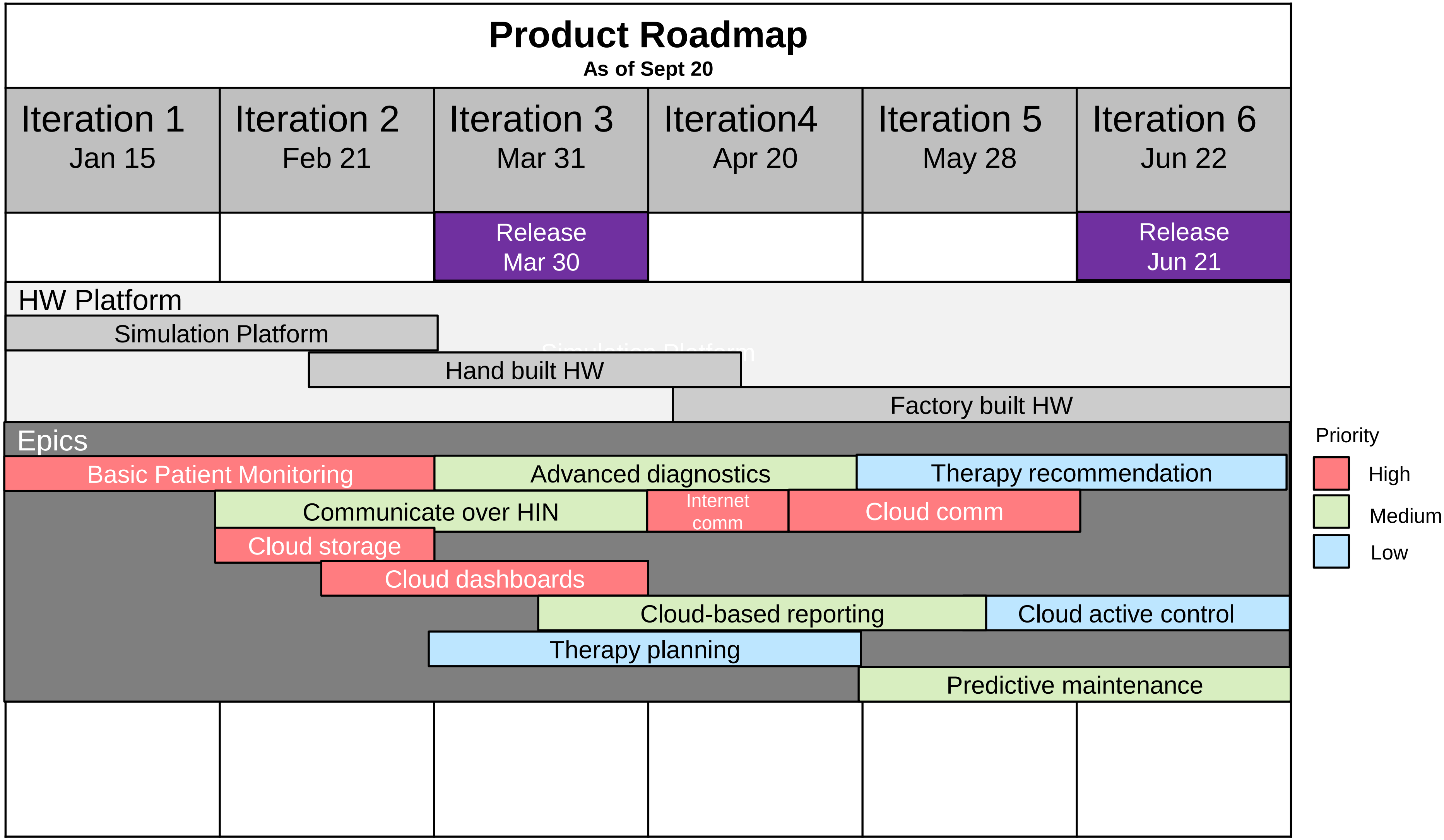
3-9 months

Contents

Product backlog, features, and user stories

© 2019 Bruce Powel Douglass, Ph.D.

Time-Based Product Roadmap



Product Roadmap

As of Sept 20

Iteration 1 Jan 15	Iteration 2 Feb 21	Iteration 3 Mar 31	Iteration4 Apr 20	Iteration 5 May 28	Iteration 6 Jun 22
		Release Mar 30			Release Jun 21

HW Platform

- Simulation Platform
- Hand built HW
- Factory built HW

Epics

- Basic Patient Monitoring
- Advanced diagnostics
- Therapy recommendation
- Communicate over HIN
- Internet comm
- Cloud comm
- Cloud storage
- Cloud dashboards
- Cloud-based reporting
- Cloud active control
- Therapy planning
- Predictive maintenance

Priority

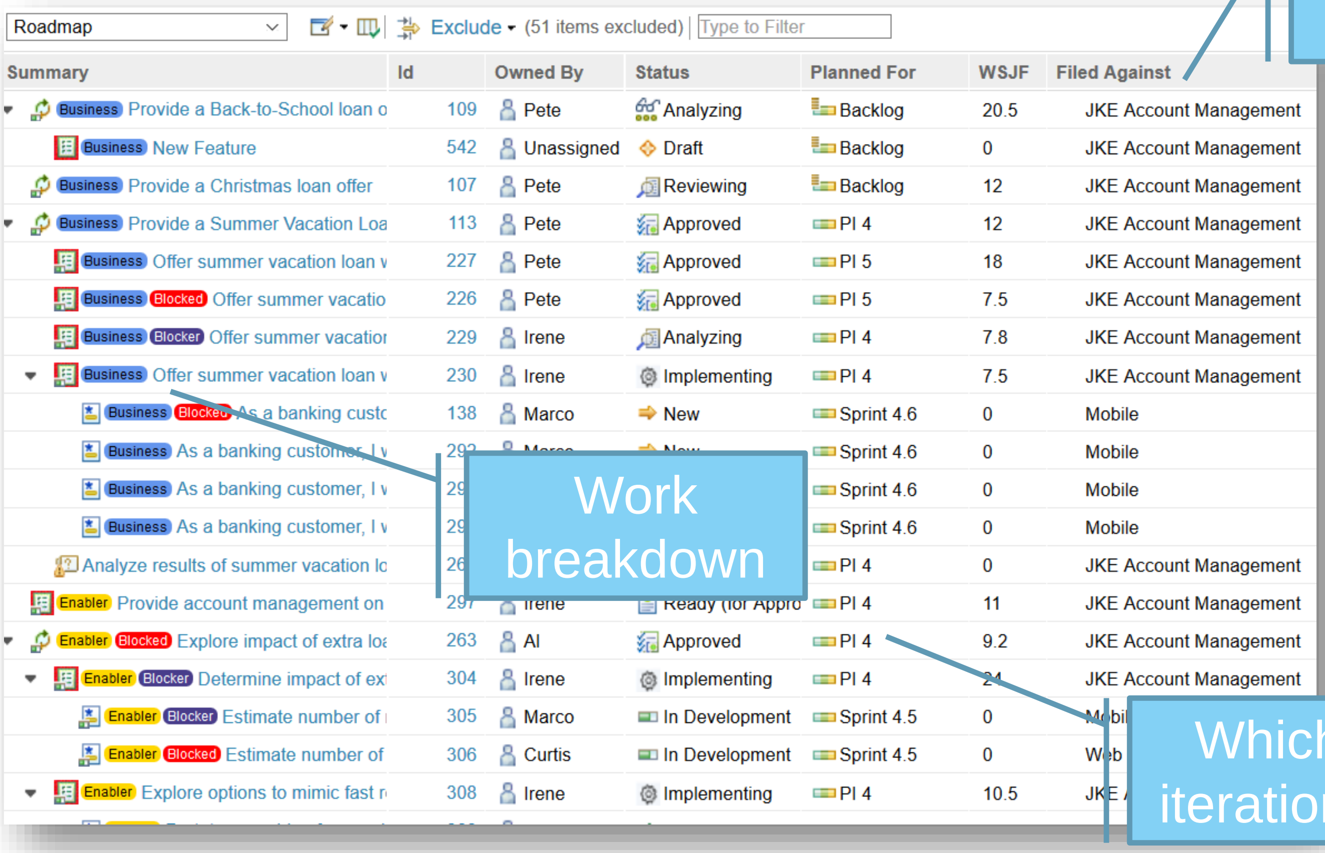
🟥 High

🟩 Medium

🟦 Low

© 2019 Bruce Powel Douglass, Ph.D.

Roadmaps in Rational Team Concert



Roadmap | Exclude (51 items excluded) | Type to Filter

Summary	Id	Owned By	Status	Planned For	WSJF	Filed Against
Business Provide a Back-to-School loan o	109	Pete	Analyzing	Backlog	20.5	JKE Account Management
Business New Feature	542	Unassigned	Draft	Backlog	0	JKE Account Management
Business Provide a Christmas loan offer	107	Pete	Reviewing	Backlog	12	JKE Account Management
Business Provide a Summer Vacation Loa	113	Pete	Approved	PI 4	12	JKE Account Management
Business Offer summer vacation loan v	227	Pete	Approved	PI 5	18	JKE Account Management
Business Blocked Offer summer vacatio	226	Pete	Approved	PI 5	7.5	JKE Account Management
Business Blocker Offer summer vacatior	229	Irene	Analyzing	PI 4	7.8	JKE Account Management
Business Offer summer vacation loan v	230	Irene	Implementing	PI 4	7.5	JKE Account Management
Business Blocked As a banking custc	138	Marco	New	Sprint 4.6	0	Mobile
Business As a banking customer, I v	292	Marco	New	Sprint 4.6	0	Mobile
Business As a banking customer, I v	29			Sprint 4.6	0	Mobile
Business As a banking customer, I v	29			Sprint 4.6	0	Mobile
Analyze results of summer vacation lo	26			PI 4	0	JKE Account Management
Enabler Provide account management on	297	Irene	Ready (for Appro	PI 4	11	JKE Account Management
Enabler Blocked Explore impact of extra loa	263	Al	Approved	PI 4	9.2	JKE Account Management
Enabler Blocker Determine impact of ext	304	Irene	Implementing	PI 4	24	JKE Account Management
Enabler Blocker Estimate number of	305	Marco	In Development	Sprint 4.5	0	Mobil
Enabler Blocked Estimate number of	306	Curtis	In Development	Sprint 4.5	0	Web
Enabler Explore options to mimic fast r	308	Irene	Implementing	PI 4	10.5	JKE A

Which team?

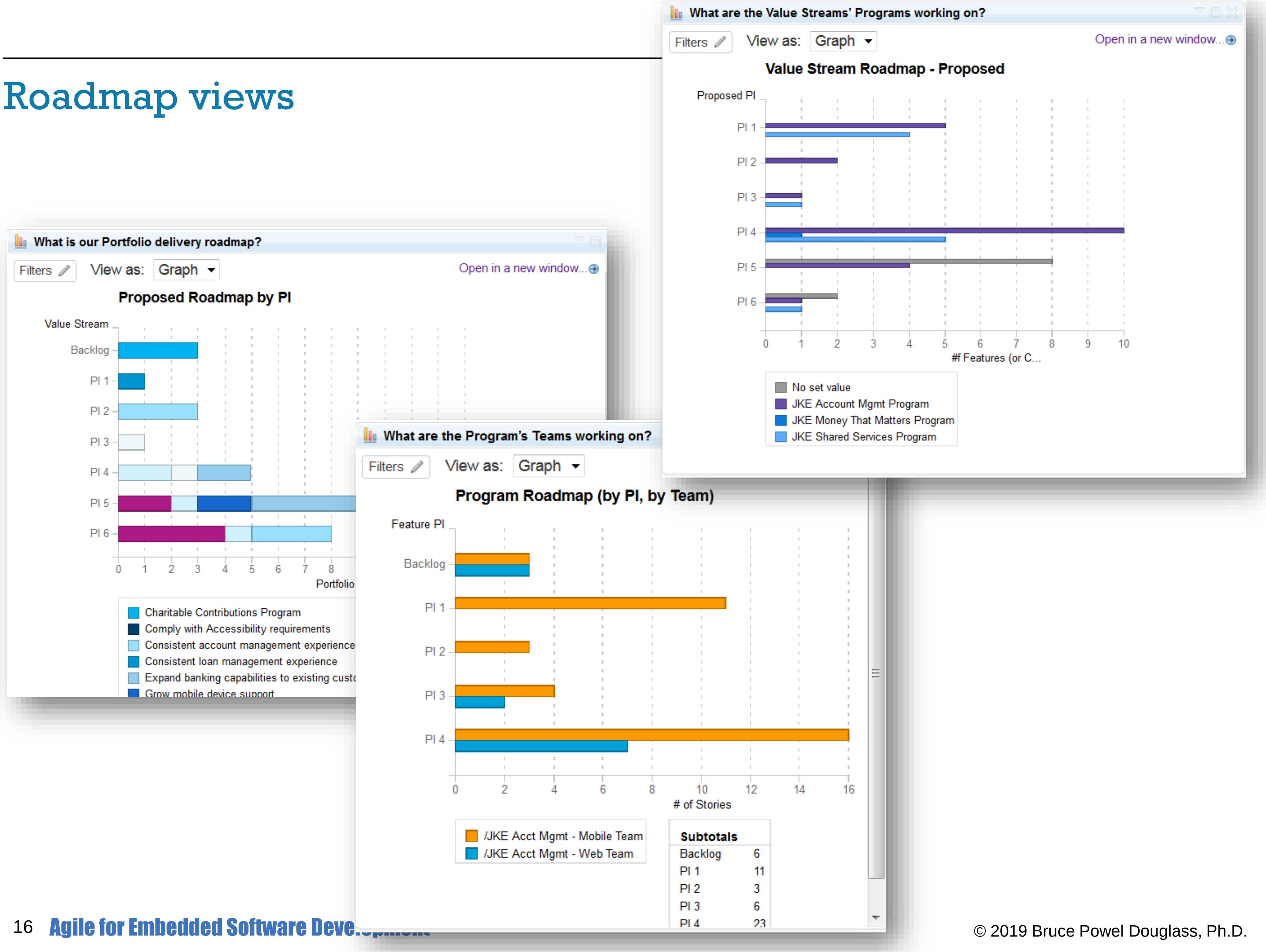
Work breakdown

Which iteration?

WSJF = Weighted Shortest Job First

© 2019 Bruce Powel Douglass, Ph.D.

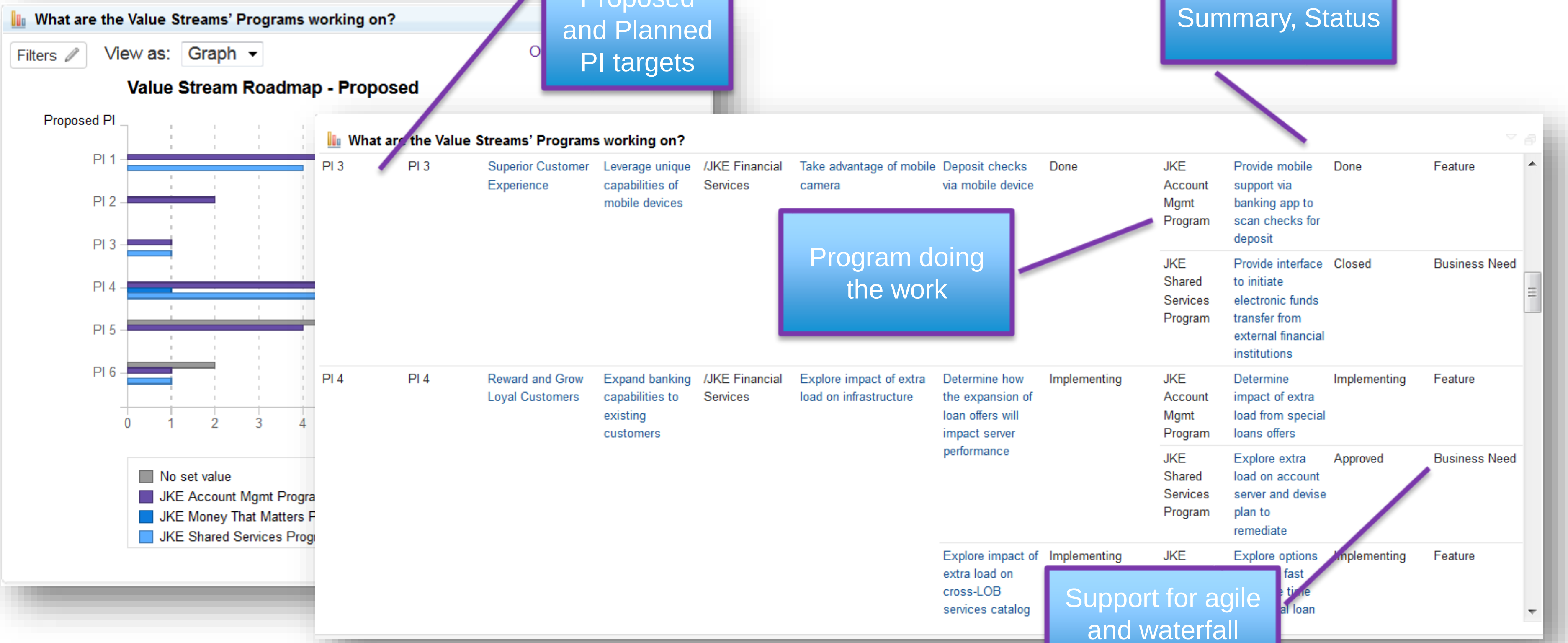
Roadmap views


What are the Value Streams' Programs working on?
Filters
View as: Graph
Open in a new window...
Value Stream Roadmap - Proposed
Proposed PI
PI 1
PI 2
PI 3
PI 4
PI 5
PI 6
0
1
2
3
4
5
6
7
8
9
10
## Features (or C...
No set value
JKE Account Mgmt Program
JKE Money That Matters Program
JKE Shared Services Program
What is our Portfolio delivery roadmap?
Filters
View as: Graph
Open in a new window...
Proposed Roadmap by PI
Value Stream
Backlog
PI 1
PI 2
PI 3
PI 4
PI 5
PI 6
0
1
2
3
4
5
6
7
8
Portfolio
Charitable Contributions Program
Comply with Accessibility requirements
Consistent account management experience
Consistent loan management experience
Expand banking capabilities to existing custo
Grow mobile device support
What are the Program's Teams working on?
Filters
View as: Graph
Program Roadmap (by PI, by Team)
Feature PI
Backlog
PI 1
PI 2
PI 3
PI 4
0
2
4
6
8
10
12
14
16
# of Stories
/JKE Acct Mgmt - Mobile Team
/JKE Acct Mgmt - Web Team

Subtotals	
Backlog	6
PI 1	11
PI 2	3
PI 3	6
PI 4	23

© 2019 Bruce Powel Douglass, Ph.D.

Roadmaps – Drilling into the details



What are the Value Streams' Programs working on?

PI 3	PI 3	Superior Customer Experience	Leverage unique capabilities of mobile devices	/JKE Financial Services	Take advantage of mobile camera	Deposit checks via mobile device	Done	JKE Account Mgmt Program	Provide mobile support via banking app to scan checks for deposit	Done	Feature
								JKE Shared Services Program	Provide interface to initiate electronic funds transfer from external financial institutions	Closed	Business Need
PI 4	PI 4	Reward and Grow Loyal Customers	Expand banking capabilities to existing customers	/JKE Financial Services	Explore impact of extra load on infrastructure	Determine how the expansion of loan offers will impact server performance	Implementing	JKE Account Mgmt Program	Determine impact of extra load from special loans offers	Implementing	Feature
								JKE Shared Services Program	Explore extra load on account server and devise plan to remediate	Approved	Business Need
						Explore impact of extra load on cross-LOB services catalog	Implementing	JKE	Explore options [illegible] fast [illegible] time [illegible] loan	Implementing	Feature

Proposed and Planned PI targets

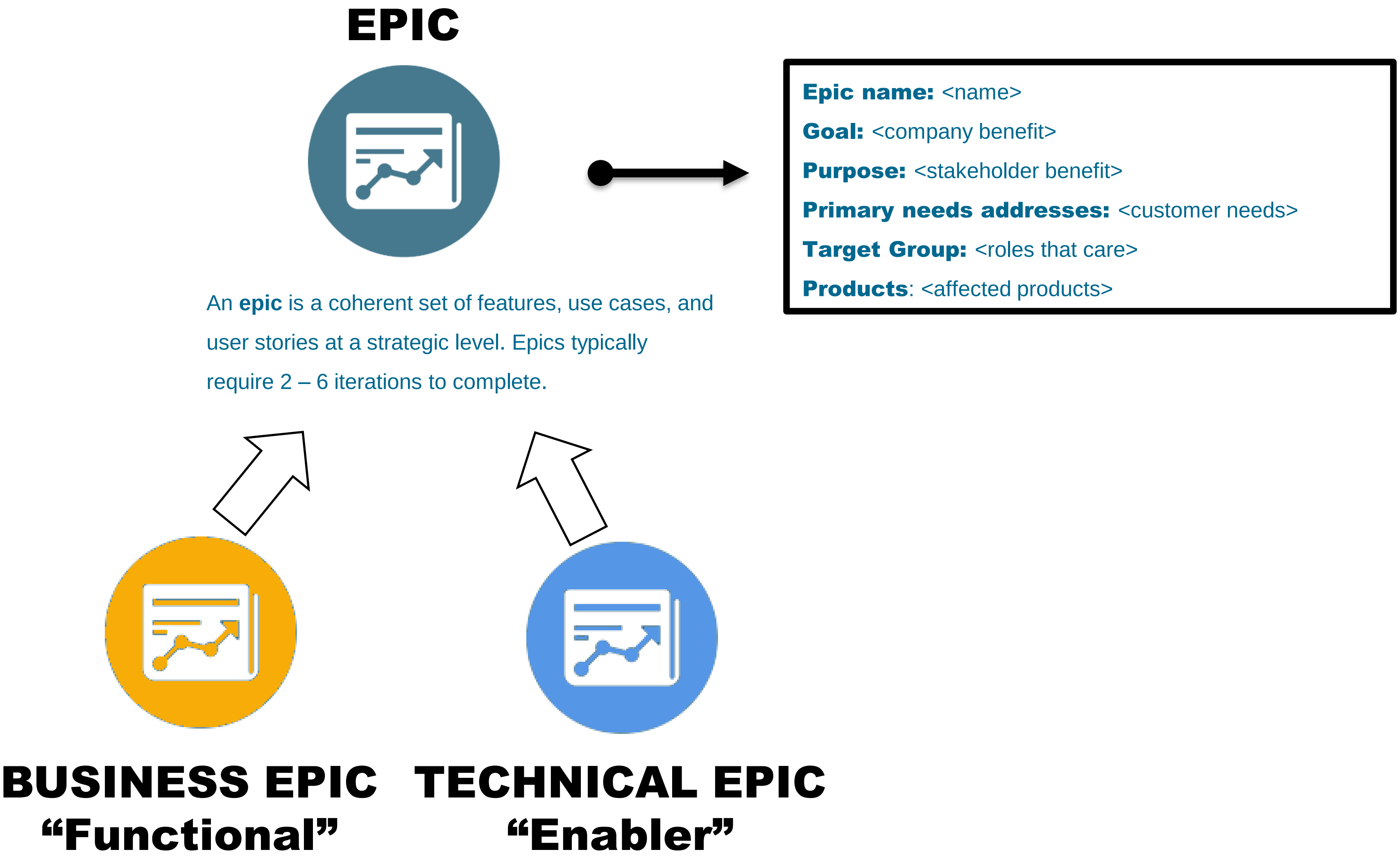
Program work: Summary, Status

Program doing the work

Support for agile and waterfall Programs

© 2019 Bruce Powel Douglass, Ph.D.

Defining Epics



EPIC

Epic name: <name>

Goal: <company benefit>

Purpose: <stakeholder benefit>

Primary needs addresses: <customer needs>

Target Group: <roles that care>

Products: <affected products>

An **epic** is a coherent set of features, use cases, and user stories at a strategic level. Epics typically require 2 – 6 iterations to complete.

BUSINESS EPIC
“Functional”

TECHNICAL EPIC
“Enabler”

© 2019 Bruce Powel Douglass, Ph.D.

Example Epic

Medical Device

Epic name: Surgical ventilation

Goal: Establish company in the high-end surgical medical device market. Also reduce the number of different ventilation architecture platforms supported by the company by creating a customizable device.

Purpose: Provide ventilation which is highly reliable, easy to configure, easy to maintain, and interacts with the HIN

Primary needs addresses:

- Simplify set up time
- Provide highly reliable ventilation even during patient episodes and loss of power
- Tie in reporting to hospital information network (HIN)

Target Group: Surgical anesthesiologist

Products: Mixologist series of ventilators, Merlin ventilator

Navigation Device

Epic name: Satellite navigation

Goal: Enter into the market for aircraft and ground craft manual and autonomous vehicle navigation

Purpose: Provide navigation that can use GPS, BDS, and GLASNOS satellite systems for global navigation. Can provide displays for manual navigation control or digital information for autonomous control.

Primary needs addresses:

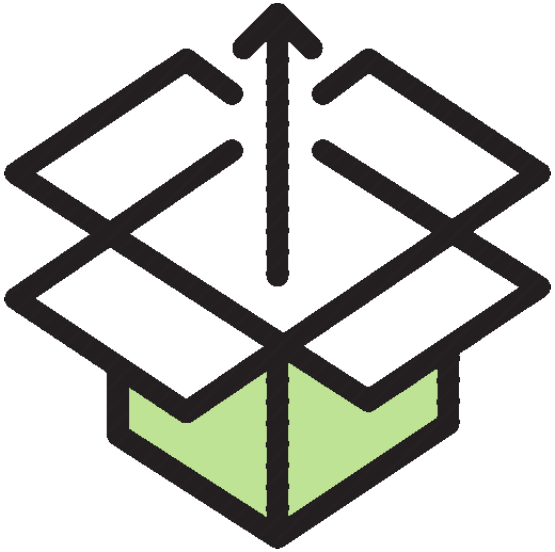
- Short configuration time (<10s)
- Highly accurate position and velocity in 3 dimensions
- Short lag (<50ms)
- Pre-certified for DO-178 and ISO 26262
- Can provide highly detailed map display
- Can provide digital coordinate and velocity data for autonomous navigation

Target Group: Military and commercial aircraft and ground craft integrators

Products: Pathfinder

© 2019 Bruce Powel Douglass, Ph.D.

Agile Release Plan



01

Purpose

The goal of the **Release Plan** is to show how the product backlog maps to the set of iterations and releases, especially in the near term.

02

Description

The release plan is a high level plan for a set of iteration and reflects expectations about when various product features will be released. The release plan is used to decide whether or not the project will produce enough ROI to at least pay for itself, and therefore whether or not we should proceed.

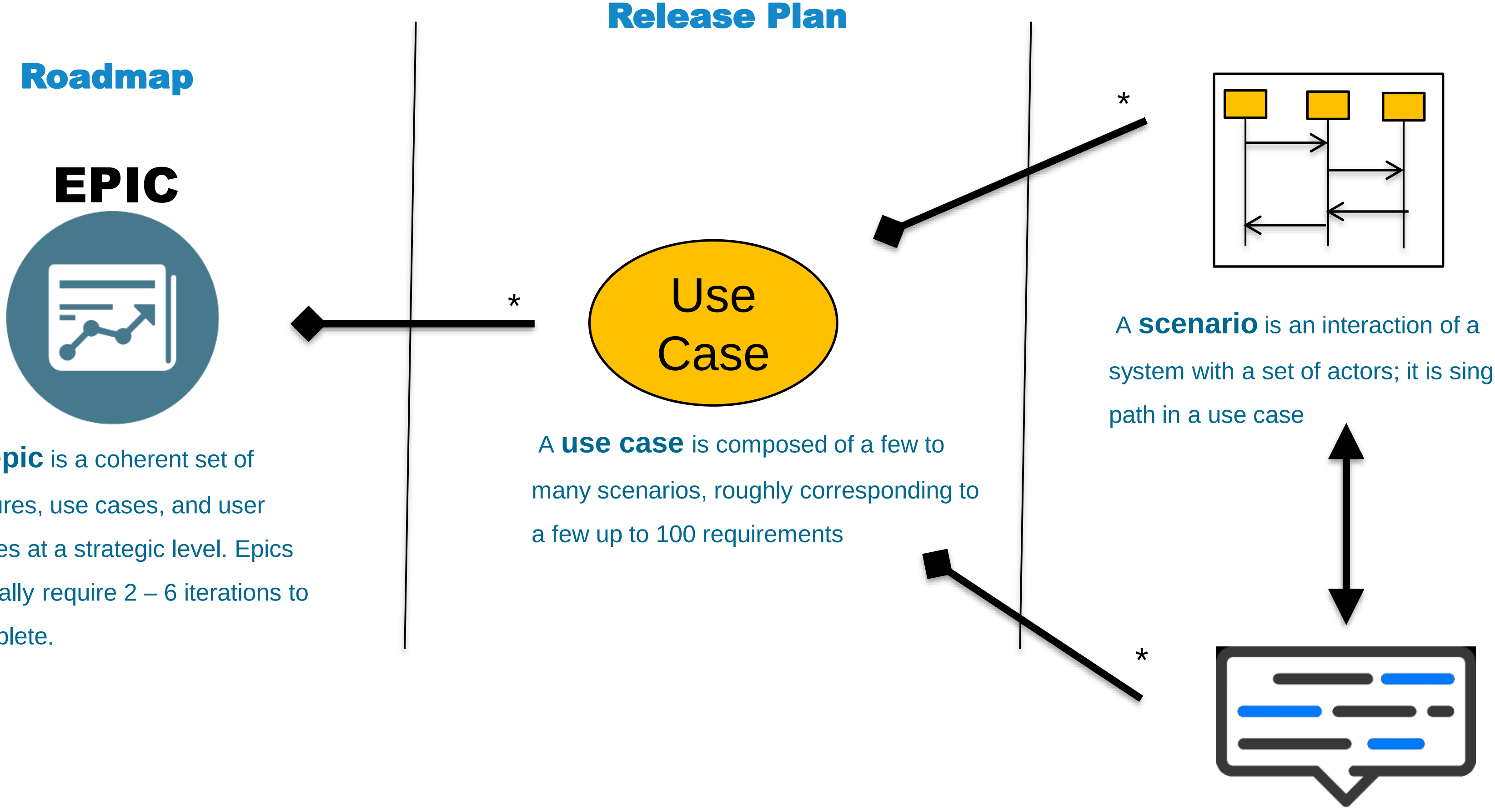
03

Hints

1. Start by breaking down epics into use case and features (and if necessary, those into user stories). Be more detailed for near term than far-term iterations
2. Release plan is updated continuously – or at least very frequently.
3. Note that an iteration produces an **increment** which is a potential release, while a **release** is delivered to relevant stakeholders.
4. High priority epics and user stories are developed in earlier iterations than those of low priority
5. It is usually better to put use cases (or features) in a release plan than stories, as the latter may be too granular
6. Care should be taken to ensure that the use cases can be delivered within a single iteration
7. It is at this time that the **Risk Plan** should be done as well

© 2019 Bruce Powel Douglass, Ph.D.

Epic Use Cases



A **scenario** is an interaction of a system with a set of actors; it is single path in a use case

A **use case** is composed of a few to many scenarios, roughly corresponding to a few up to 100 requirements

An **epic** is a coherent set of features, use cases, and user stories at a strategic level. Epics typically require 2 – 6 iterations to complete.

A **user story** is a single interaction of one or more actors with the product to achieve a goal.

© 2019 Bruce Powel Douglass, Ph.D.

Epic Use Cases Example

Epic name: Surgical ventilation

Goal: Establish company in the high-end surgical medical device market. Also reduce the number of different ventilation architecture platforms supported by the company by creating a customizable device.

Purpose: Provide ventilation which is highly reliable, easy to configure, easy to maintain, and interacts with the

Primary needs addresses:

- Simplify set up time
- Provide highly reliable ventilation even during patient episodes and loss of power
- Tie in reporting to hospital information network (HIN)

Target Group: Surgical anesthesiologist

Products: Mixologist series of ventilators, Merlin ventil

Use Case

Use case name: Mix Gases

Purpose: Allow accurate mixing of gases for delivery

Description: Provides the well-controlled mixing of up to 6 different gases from wall supplies

Actors: Gas supply, breathing circuit, physician

Pre-conditions: Gas is available, system is connected to breathing circuit

Post-conditions: mixed gas is delivered at the percentages and rates commanded

Constraints: total output flow is limits to 100 L/min

Risks: None

Use Case Points: 10

Use Case

Use case name: Monitor device health

Purpose: Identify system failures that could lead to patient episode

Description: The system monitors actuators and sensors to ensure that they are operating properly.

Actors: physician

Pre-conditions: system is on and has initial POST

Post-conditions: Errors are logged and reported to attending physician

Constraints: none

Risks: It may not be possible to identify gas leaks

Use Case Points: 6

Use Case

Use case name: CO2 Scavenging

Purpose: Remove CO2 from the expiratory gas

Description: Removes almost all expired CO2 from expired gas but alarms if CO2 exceeds threshold.

Actors: Breathing circuit

Pre-conditions: Connected to the breathing circuit

Post-conditions: Removes CO2 or alerts attending physician

Constraints: expiratory flow is limited to 100 L/min max

Risks: Unsure if we can meet the target CO2 concentration at high flow rate

Use Case Points: 4

Use Case

Use case name: Monitor patient parameters

Purpose: Provide the physician with timely information about patient health

Description: Monitors and reports SpO2, O2 input flow, O2 input percentage, heart rate, and NIBP

Actors: Physician

Pre-conditions: system is on and has initial POST

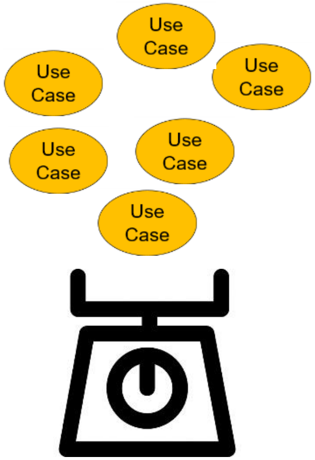
Post-conditions: patient data displayed in a timely fashion

Constraints: none

Risks: Our current SpO2 OEM vendor is going out of business and it isn't clear there is a **viable** replacement

Use Case Points: 7

Estimating use case / user story size



01

Purpose

Size/work effort estimation is important because it allows us to allocate work to iterations with some confidence of being able to achieve the work.

02

Description

Use case points are a common agile technique using approximate relative, rather than absolute sizing.

While estimating use case points, we assign a point value to each use case. Relative values are more important than the raw values. A 4-pt use case would take 4 times most effort to create than a 1-pt use case.

Alternatively, an absolute measure, such as work hours can use used as an estimate.

03

Hints

Use case points are not generally a continuous range. It is common to use doubled numbers such as 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100 or Fibonacci sequence: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

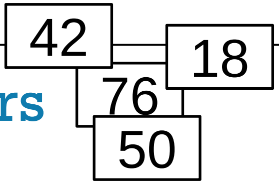
04

Procedure

1. List all use cases, risk spikes, and technical work items
2. Arrange them in order of effort, from smallest to largest
3. Sort them until consensus is reached on the ordering
4. That done, then assign point values to the use cases, starting at the bottom (smallest)
5. As you move up the list, size each based on its relative size of the one below

© 2019 Bruce Powel Douglass, Ph.D.

Points vs Hours



- Use case or story points
  - Are a relative size estimation technique
  - Avoid the problem of being overly specific about effort estimation when information is unavailable to accurately estimate work effort
  - The higher the level the work item, the more sense relative estimating makes
    - Use cases and stories are often estimated with points
    - Work tasks are usually estimated with hours
- **Velocity** is the rate at which points are delivered. On early iterations, this is estimated, but the accuracy of velocity is improved in later iterations because we can calibrate it based on actual evidence.
- Velocity varies based on
  - Team size
  - Team skill
  - Domain knowledge
  - Use case complexity
  - Tools and technology (e.g. automation)
  - Development environment factors
  - Regulation factors

© 2019 Bruce Powel Douglass, Ph.D.

Planning Poker

01

Purpose

Planning poker is a quick and easy design game for estimating effort for work items.

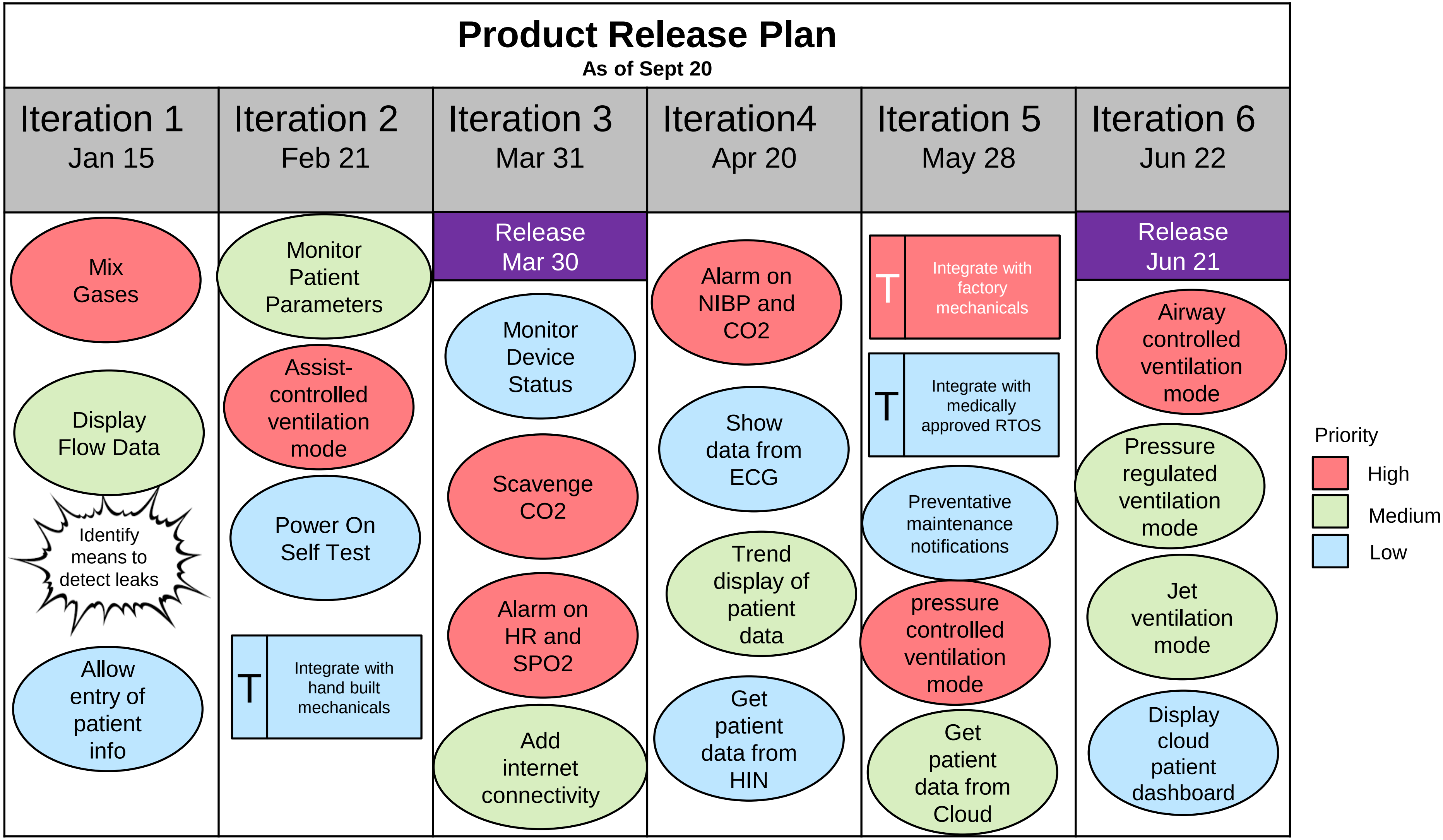
02

Description

1. The moderator will also need to prepare the list of use cases to size, and a set of planning cards to provide to each player.
   - The number of cards in this set depends on the number of estimating categories. Commonly the cards have values like 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100
2. Estimation is performed:
   1. The estimators discuss the feature, asking questions of the product owner as needed.
   2. Each estimator privately selects one card to represent his or her estimate.
   3. When all participants have made their choice, all the cards are turned over at the same time.
   4. If all estimators selected the same value, that becomes the estimate. If not, the estimators discuss their estimates. The high and low estimators should especially share their reasons.
   5. Repeat this process until consensus is achieved or it is decided that more information is required.
3. Repeat for all items to be estimated

© 2019 Bruce Powel Douglass, Ph.D.

Time-Based Product Release Plan



Product Release Plan

As of Sept 20

Iteration 1 Jan 15	Iteration 2 Feb 21	Iteration 3 Mar 31	Iteration4 Apr 20	Iteration 5 May 28	Iteration 6 Jun 22
		Release Mar 30			Release Jun 21
🟥 Mix Gases	🟩 Monitor Patient Parameters	🟦 Monitor Device Status	🟥 Alarm on NIBP and CO2	🟥 T Integrate with factory mechanicals	🟥 Airway controlled ventilation mode
🟩 Display Flow Data	🟥 Assist-controlled ventilation mode	🟥 Scavenge CO2	🟦 Show data from ECG	🟦 T Integrate with medically approved RTOS	🟩 Pressure regulated ventilation mode
Identify means to detect leaks	🟦 Power On Self Test	🟥 Alarm on HR and SPO2	🟩 Trend display of patient data	🟦 Preventative maintenance notifications	🟩 Jet ventilation mode
🟦 Allow entry of patient info	🟦 T Integrate with hand built mechanicals	🟩 Add internet connectivity	🟦 Get patient data from HIN	🟥 pressure controlled ventilation mode	🟦 Display cloud patient dashboard
				🟩 Get patient data from Cloud	

Priority

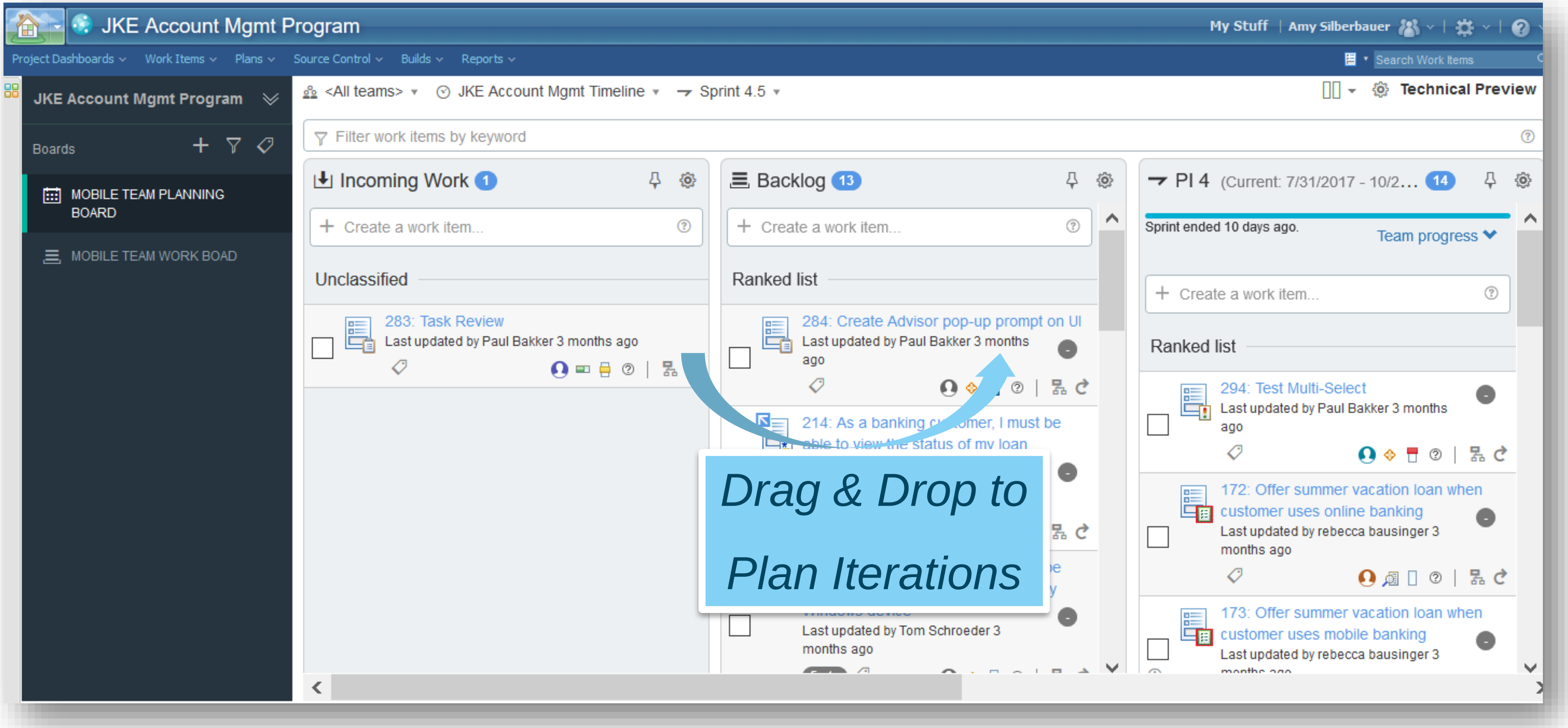
🟥 High

🟩 Medium

🟦 Low

© 2019 Bruce Powel Douglass, Ph.D.

Release Plan in Team Concert



283: Task Review
Last updated by Paul Bakker 3 months ago

284: Create Advisor pop-up prompt on UI
Last updated by Paul Bakker 3 months ago

214: As a banking customer, I must be able to view the status of my loan

Last updated by Tom Schroeder 3 months ago

Sprint ended 10 days ago.

294: Test Multi-Select
Last updated by Paul Bakker 3 months ago

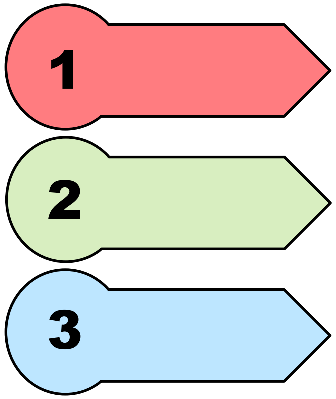
172: Offer summer vacation loan when customer uses online banking
Last updated by rebecca bausinger 3 months ago

173: Offer summer vacation loan when customer uses mobile banking
Last updated by rebecca bausinger 3 months ago

Drag & Drop to Plan Iterations

© 2019 Bruce Powel Douglass, Ph.D.

Use Case Prioritization



01

Purpose

A key aspect of release planning is determining what work items should be done when. This is known as *prioritization.*

02

Description

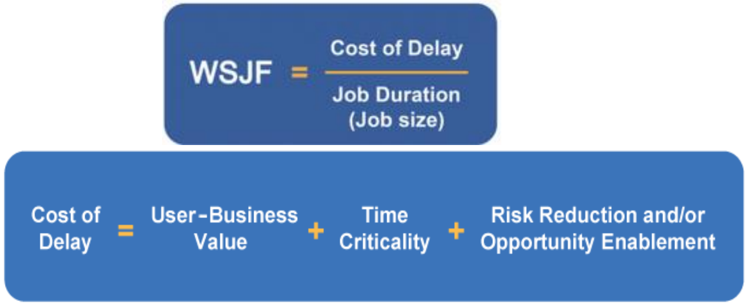
Priority is used to order a set of work tasks with the basic rule that higher priority items should be worked before lower priority items. In some systems a lower numeric value indicates a higher priority, while in others, the rule is reversed. *I recommend lower number == higher priority*. Priority can be used to optimize work efforts using a number of different criteria, all potentially simultaneously.

03

Criteria

1. Cost of Delay, which includes
   1. Criticality – how important it is to the product success
   2. Usefulness - how useful it is to the user
   3. Urgency - when customer needs it
   4. Risk – reduction or project risk or opportunity loss
2. Difficulty/ Time/ Effort – how hard is it to develop?
3. Sensical sequencing
4. Dependency (on other features, infrastructure or hardware/platform capability)
5. Congruency to the theme of the iteration, including the features of the hardware being co-developed
6. Availability of necessary resources - such as subject matter expert or specialized equipment for development, verification, or validation

Weighted Shortest Job First Prioritization



WSJF = Cost of Delay / Job Duration (Job size)

Cost of Delay = User-Business Value + Time Criticality + Risk Reduction and/or Opportunity Enablement

© 2019 Bruce Powel Douglass, Ph.D.

Prioritization with MoSCoW Method

- **MoSCoW** analysis is the prioritization technique that is recommended for business analysts in the IIBA BABOK and originating from the DSDM (dynamic software development method). According to this method, a list of use case, user stories, requirements or other work items should be categorized into the following 4 groups:

M **M: Must.** Describes a requirement that must be satisfied in the final solution for the solution to be considered a success.

S **S: Should**. Represents a high-priority item that should be included in the solution if it is possible. This is often a critical requirement but one which can be satisfied in other ways if absolutely necessary.

C **C: Could.** Describes a requirement which is considered desirable but not necessary. This will be included if time and resources permit.

W **Won't.** Represents a requirement that stakeholders have agreed will not be implemented in a given release, but may be considered in the future.

BABOK Guide - IIBA | International Institute of Business Analysis
www.iiba.org/babok-guide.aspx

A Guide to the Business Analysis Body of Knowledge (*BABOK* Guide).

Priority Poker

01

Purpose

Priority poker is a quick and easy design game for prioritizing items.

It's called priority poker because it's very similar to planning poker.

02

Description

1. The moderator will also need to prepare the list of tasks to prioritize, and a set of priority cards to provide to each player.
   - The number of cards in this set depends on how many levels of priority are useful to use in this particular case. It could be 3,5, or even 10-level scale.
2. Gather all the people that need to be involved in the prioritization process, such as stakeholders, product owners, designers, developers, domain experts and perhaps even users.
3. Prioritization is performed:
   1. The moderator reads the work item, such as a use case.
   2. Each participant chooses the card that they think is the most appropriate ranking for that task and places the card face down on the table.
   3. When all participants have made their choice, all the cards are turned over at the same time.
   4. The differences are discussed and the game goes on until the priorities are all around the same level. This may involve each player discussing which criteria they used to set the priority.
4. Repeat for the next work item

© 2019 Bruce Powel Douglass, Ph.D.

Allocating work items using priority

01

Prioritize

Prioritize the work items using your selected approach and criteria

02

Allocate

Allocate work items to iterations, including

- Use cases
- Spikes
- Technical work items
- Defects in the backlog

03

Evaluate

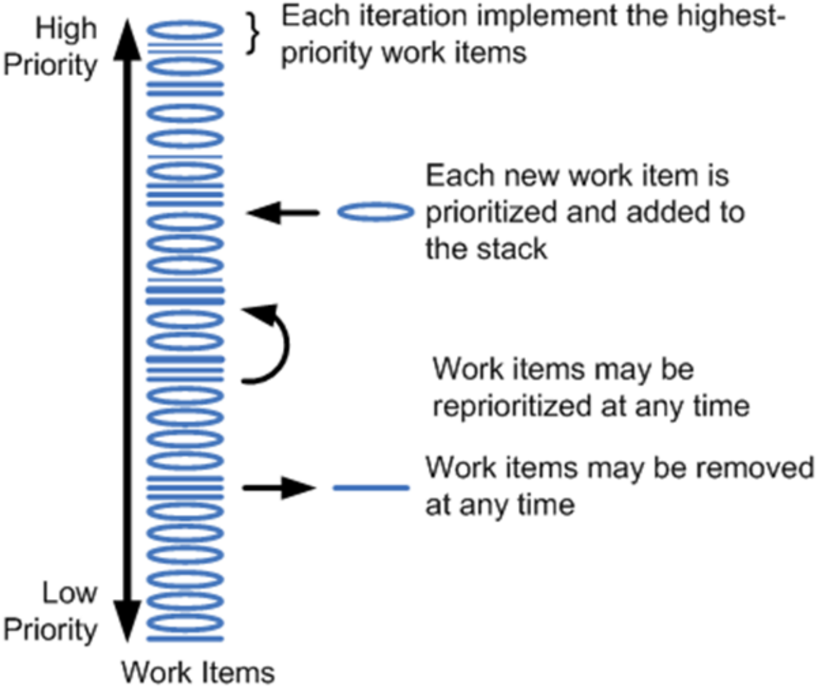
1. Does the increment trajectory make sense?
2. Is each iteration scope reasonable?

04

Revise

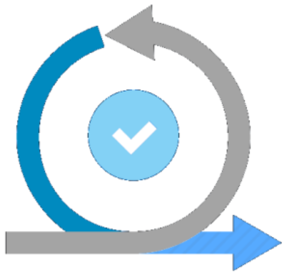
Improve the release plan by

1. Moving work items around
2. Updating the team
3. Modifying the work item scope (e.g. is a use case is too large?)


High Priority
Low Priority
Work Items
} Each iteration implement the highest-priority work items
Each new work item is prioritized and added to the stack
Work items may be reprioritized at any time
Work items may be removed at any time

© 2019 Bruce Powel Douglass, Ph.D.

Agile Iteration Plan



01

Purpose

The Iteration or Sprint Planning meeting is for team members to plan and agree on the stories or backlog items they are confident they can complete during the sprint and identify the detailed tasks and tests for delivery and acceptance

02

Description

The **Iteration Plan** focuses detailed attention on the work to be done in the upcoming sprint. The Release plan identifies the proposed set, but this is when the team evaluates it in detail and commits to the work. Primarily focused on use cases or user stories, the iteration plan also grabs spikes, planned technical work, and defects off the product backlog for iteration.

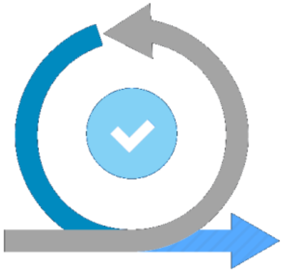
03

Hints

1. Use cases are broken down into user stories (scenarios are another way of representing user stories – as we will see)
    1. User stories are typically small 4 hr – 2 days in duration
    2. During this time, it may be discovered that a user case was woefully undersized – in this case, it can be decomposed into smaller use cases and those are prioritize and pushed back to the product backlog
    3. If the meaning or purpose of a story isn’t clear, the team can resolve this with the customer or domain expert
2. The user stores are expected to be developed using practices like
    1. Test Driven Development
    2. Continuous integration

© 2019 Bruce Powel Douglass, Ph.D.

Agile Iteration Planning



01

Gather your team

The planning session should include the product own, iteration master, and the agile development team members

02

Select work items from product backlog

Select the highest priority items in the product backlog that seem to fit within the iteration. Change its work item priority if appropriate.

03

Break use cases into user stories

For each use case, break the use case into stories (or scenarios). Each user story should be a small number of days to complete and identify related requirements.

04

Break user stories into tasks

Each user story will be performed as a set of work tasks, each of which is 4 hr – 2 days in duration.

05

Estimate work effort for tasks

The duration of each task should be estimated in work hours.

06

Put tasks into iteration backlog

Put the tasks into the backlog in priority order – highest priority to be done first.

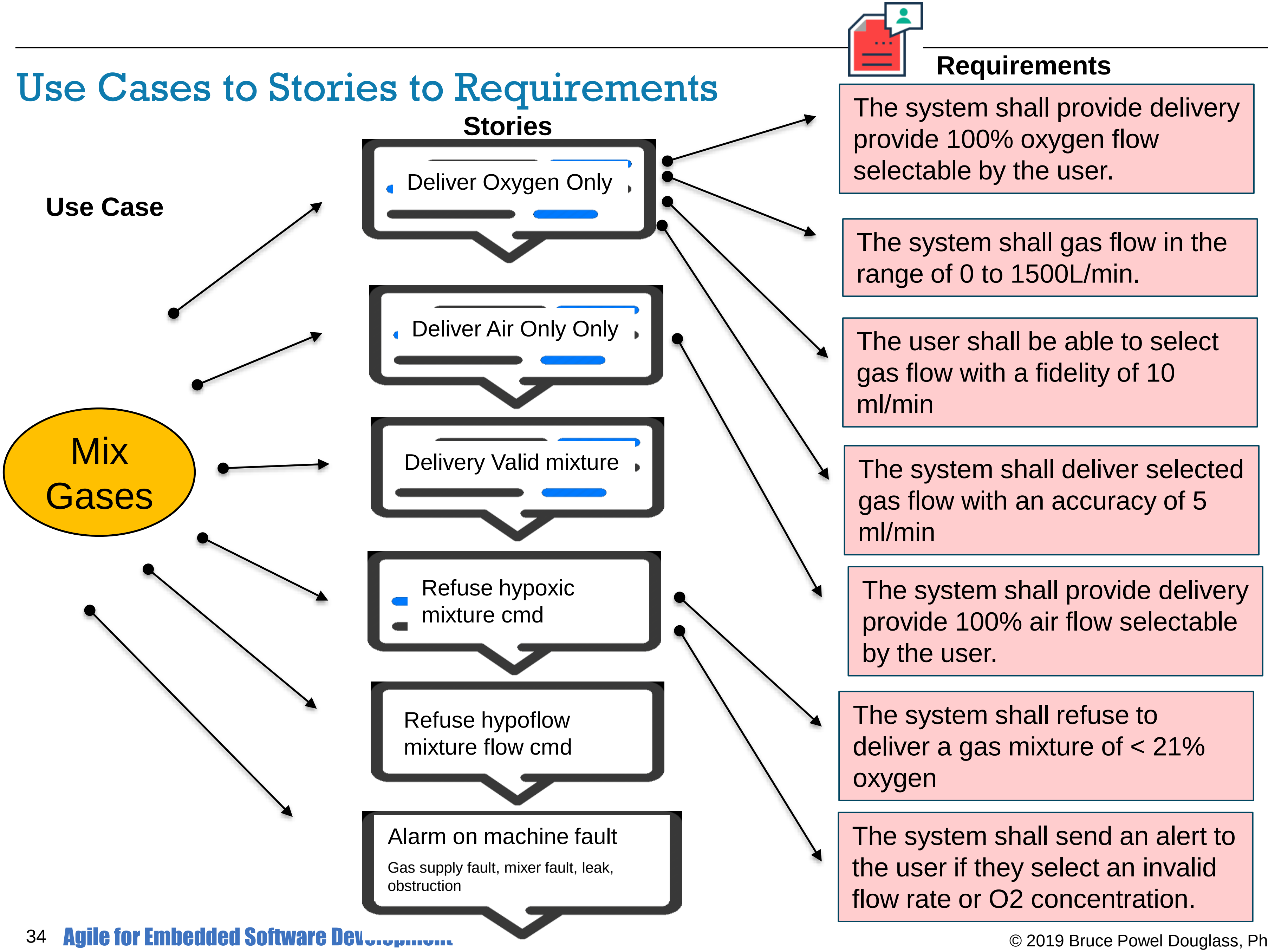
07

Evaluate team loading

Examine the work allocated to the team for the iteration and determine if the task loading is reasonable.

© 2019 Bruce Powel Douglass, Ph.D.

Use Cases to Stories to Requirements



Use Case

Mix Gases

Stories

Deliver Oxygen Only

Deliver Air Only Only

Delivery Valid mixture

Refuse hypoxic mixture cmd

Refuse hypoflow mixture flow cmd

Alarm on machine fault

Gas supply fault, mixer fault, leak, obstruction

Requirements

The system shall provide delivery provide 100% oxygen flow selectable by the user.

The system shall gas flow in the range of 0 to 1500L/min.

The user shall be able to select gas flow with a fidelity of 10 ml/min

The system shall deliver selected gas flow with an accuracy of 5 ml/min

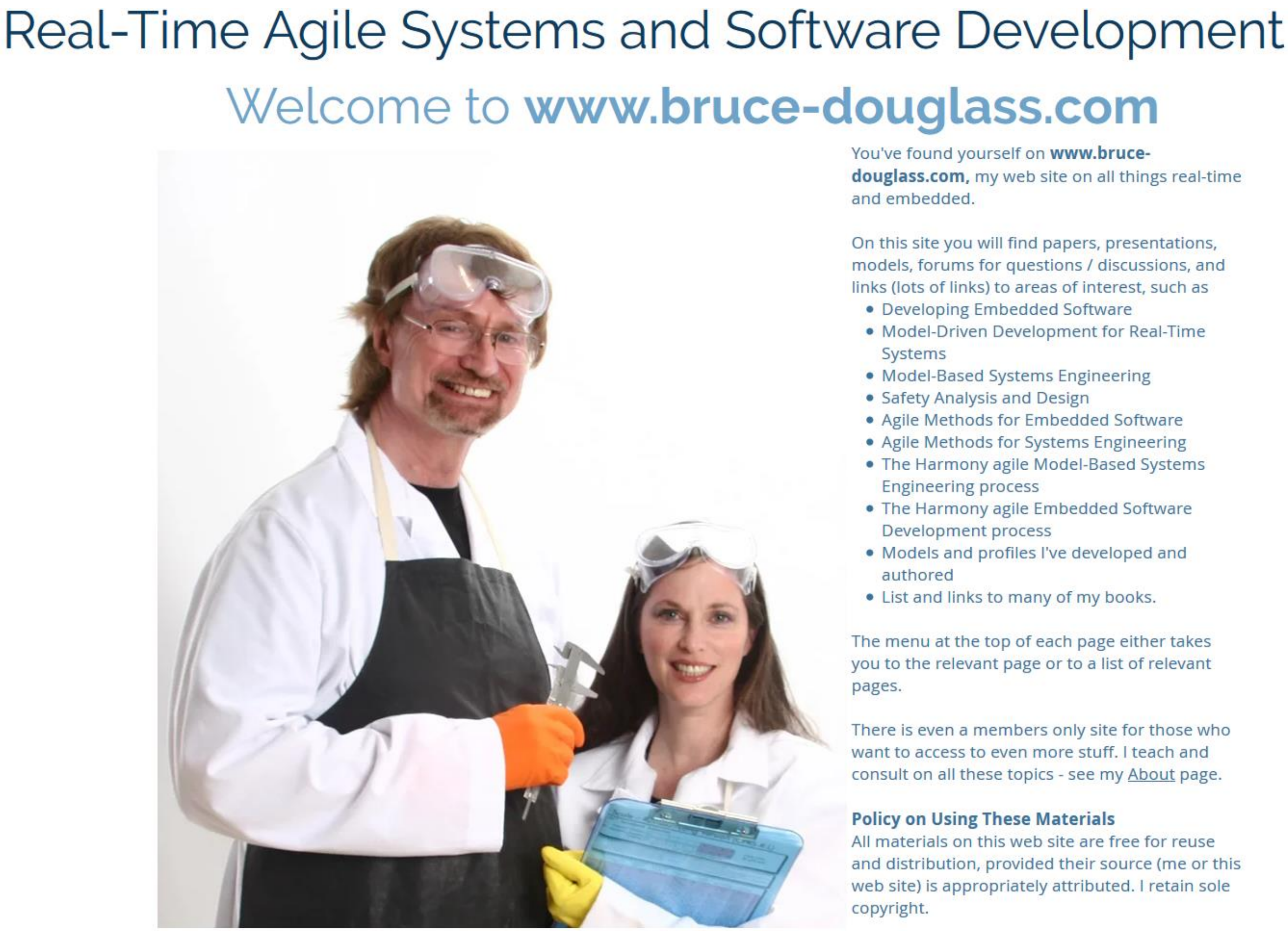
The system shall provide delivery provide 100% air flow selectable by the user.

The system shall refuse to deliver a gas mixture of < 21% oxygen

The system shall send an alert to the user if they select an invalid flow rate or O2 concentration.

© 2019 Bruce Powel Douglass, Ph.D.

Download Papers, Presentations, Models, & Profiles for Free



Real-Time Agile Systems and Software Development

Welcome to **www.bruce-douglass.com**

You've found yourself on **www.bruce-douglass.com,** my web site on all things real-time and embedded.

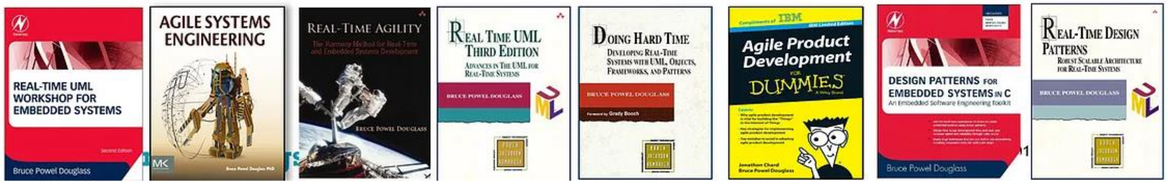
On this site you will find papers, presentations, models, forums for questions / discussions, and links (lots of links) to areas of interest, such as

- Developing Embedded Software
- Model-Driven Development for Real-Time Systems
- Model-Based Systems Engineering
- Safety Analysis and Design
- Agile Methods for Embedded Software
- Agile Methods for Systems Engineering
- The Harmony agile Model-Based Systems Engineering process
- The Harmony agile Embedded Software Development process
- Models and profiles I've developed and authored
- List and links to many of my books.

The menu at the top of each page either takes you to the relevant page or to a list of relevant pages.

There is even a members only site for those who want to access to even more stuff. I teach and consult on all these topics - see my About page.

Policy on Using These Materials
All materials on this web site are free for reuse and distribution, provided their source (me or this web site) is appropriately attributed. I retain sole copyright.


REAL-TIME UML WORKSHOP FOR EMBEDDED SYSTEMS
Bruce Powel Douglass
AGILE SYSTEMS ENGINEERING
REAL-TIME AGILITY
The Harmony Method for Real-Time and Embedded Systems Development
BRUCE POWEL DOUGLASS
REAL TIME UML THIRD EDITION
ADVANCES IN THE UML FOR REAL-TIME SYSTEMS
BRUCE POWEL DOUGLASS
DOING HARD TIME
DEVELOPING REAL-TIME SYSTEMS WITH UML, OBJECTS, FRAMEWORKS, AND PATTERNS
BRUCE POWEL DOUGLASS
Foreword by Grady Booch
IBM
Agile Product Development FOR DUMMIES
Jonathon Chard
Bruce Powel Douglass
DESIGN PATTERNS FOR EMBEDDED SYSTEMS IN C
An Embedded Software Engineering Toolkit
Bruce Powel Douglass
REAL-TIME DESIGN PATTERNS
ROBUST SCALABLE ARCHITECTURE FOR REAL-TIME SYSTEMS
BRUCE POWEL DOUGLASS

© 2019 Bruce Powel Douglass, Ph.D.