

Agile Project Management in a Waterfall World

Managing Sprints with Predictive Metrics

PMI Silicon Valley Symposia

October 8, 2013

JOHN CARTER
TCGEN INC
jcarter@tcgen.com



AGILITY CASE STUDIES - SUCCESS STORIES

Agile success stories abound...

- Shortening of internationalization of software from 11 months down to 1 month
- Reducing time to release from every 8 months down to every 2 months
- Having a savings of 40-50% of product development cycle time

But all of these successes were in SW... how can we apply to Physical Products/Systems?

Conducted research of case studies where Agile methods were applied to Products/Systems with over a dozen participants



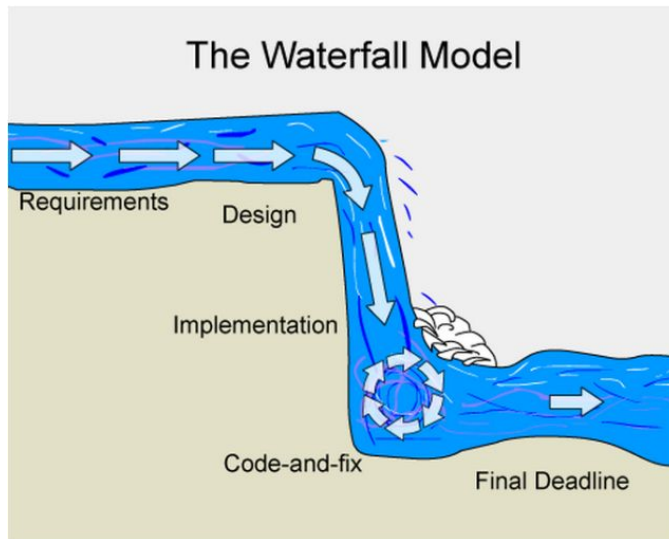
ELEVATOR PITCH

Elements that you can borrow from Agile Software Methodology that can be applied to complex hardware/software programs to increase programs speed and flexibility

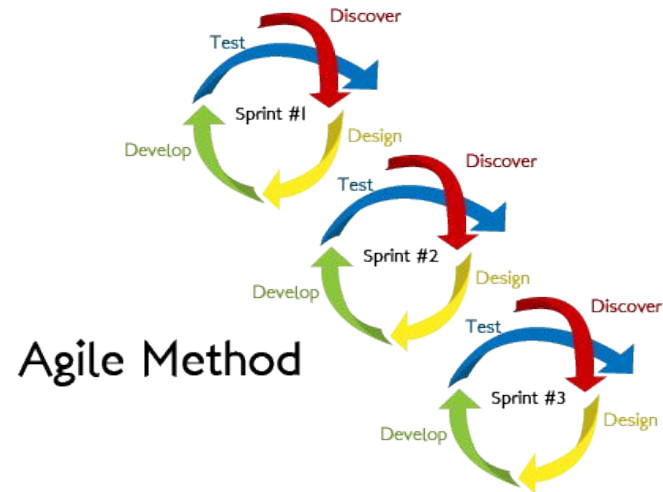
But...The challenge is how do we apply Agile Software Methods to programs that...

- Have components with long lead times?
- Development partners that do things their own way?
- Long supply chains with components, sub-assemblies, and final assemblies that need integration around the world?
- Medical products that require FDA compliance?
- Large software platforms that are developed using Waterfall Methods?

HOW IS AGILE DIFFERENT FROM WATERFALL?

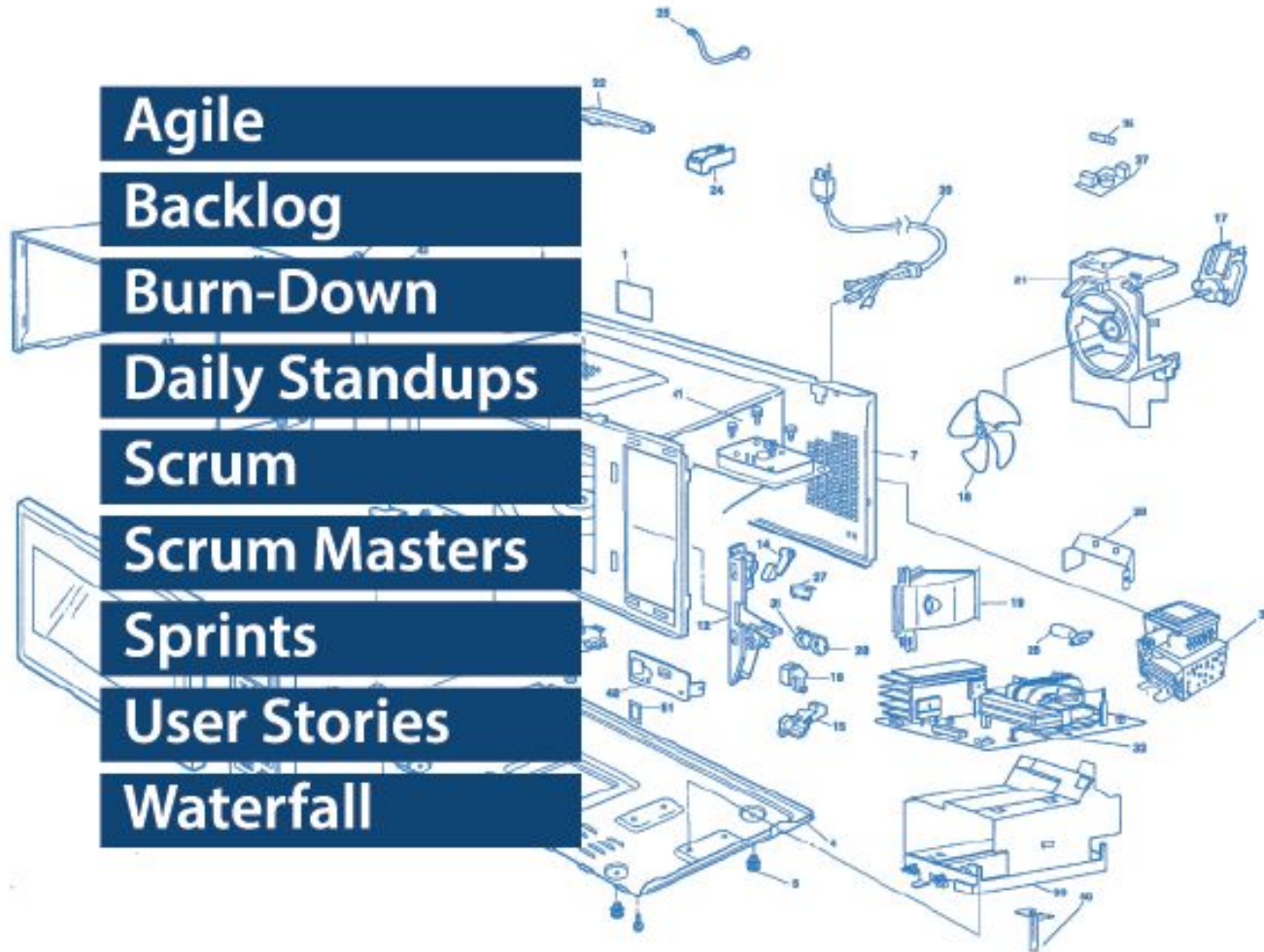


Freeze the Specs and Don't Look Back



Develop Fast, Learn, Improve

KEY AGILE & DEVELOPMENT TERMS



AGILE MANIFESTO – HARD TO DO IN SYSTEMS?

1. Continuous delivery of valuable software
2. Welcome changing requirements
3. Deliver working software frequently
4. Business people and developers work together daily
5. Projects require motivated individuals, support & trust
6. Face-to-face conversation is most efficient
7. Working software is the measure of progress
8. Agile processes promote sustainable development
9. Continuous attention to technical excellence
10. Simplicity---is essential
11. The best designs emerge from self-organizing teams
12. At regular intervals, the team reflects

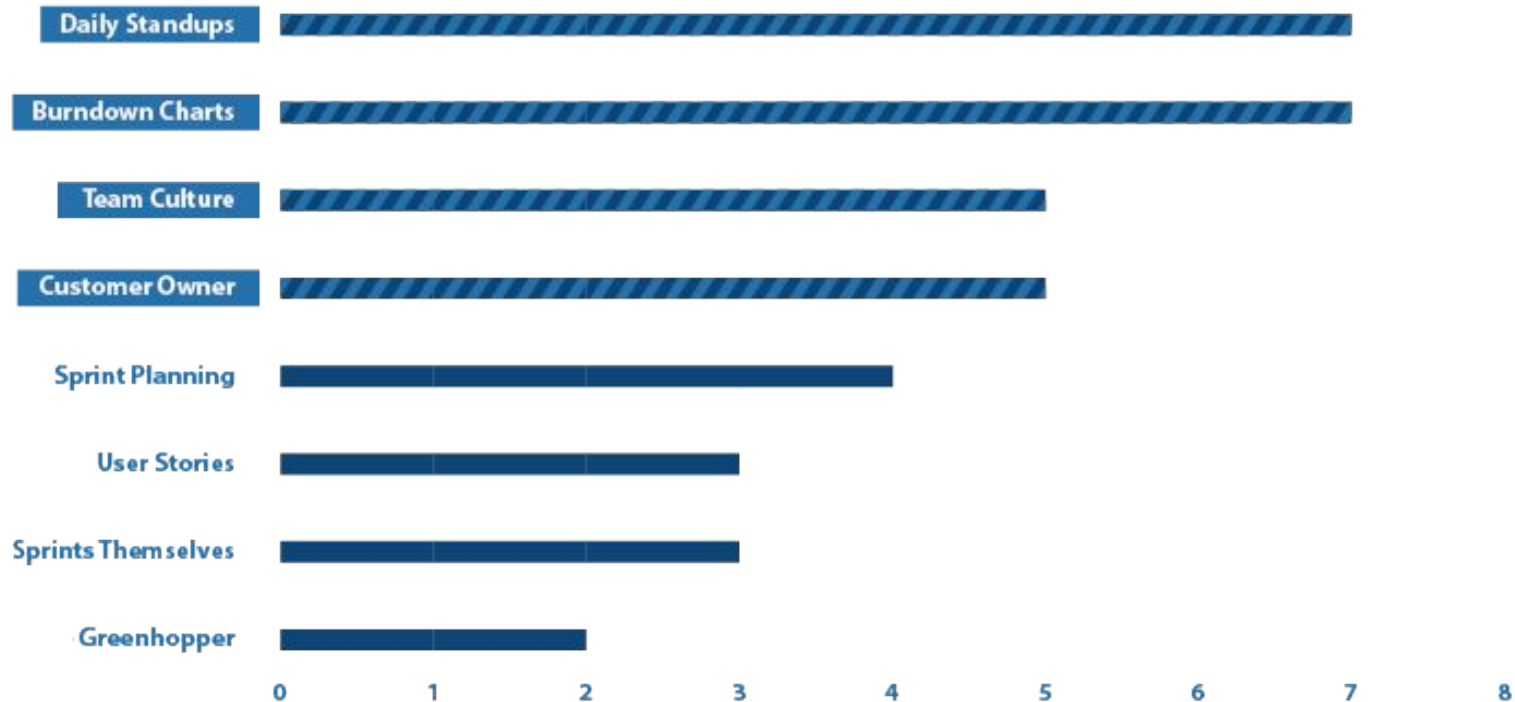
SCRUM METHODOLOGY – BEST PRACTICES

The most common implementation of the Agile Manifesto is Scrum



AGILE METHODOLOGY – BEST PRACTICES

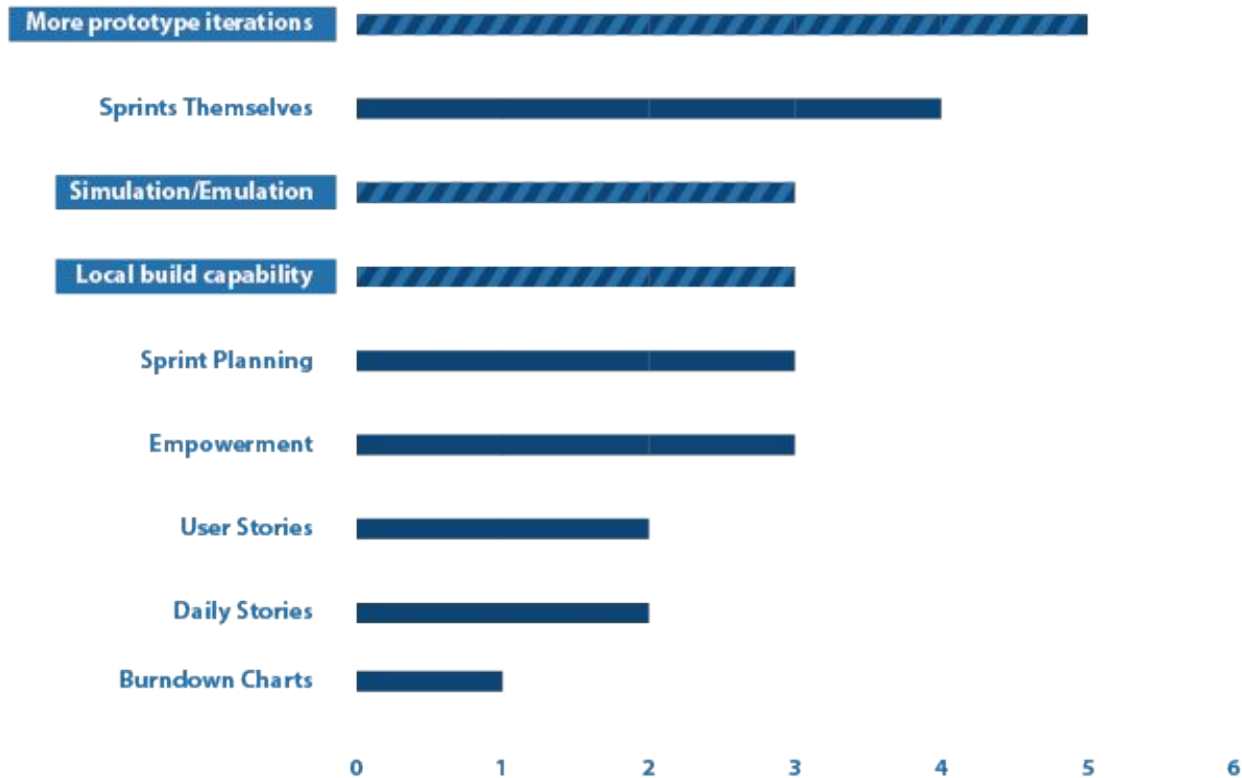
Question: What are the most impactful elements of Agile/Scrum applied to SW?



Four of the top Agile practices can be applied to Systems too

AGILE METHODOLOGY –APPLIED TO HW/SYSTEMS

Question: What are the most impactful elements of Agile/Scrum applied to Products/Systems?



Three of the top Agile practices in Systems have little to do with Scrum

So why is applying Agile/Scrum to HW/Systems so hard?

Because Agile/Scrum is hard

**It requires higher process literacy
And greater cross functional teamwork skills**

For more sophisticated teams; not for the faint of heart

FIRST, APPLY THE FOLLOWING CULTURAL ASPECTS

Adopt the concept of

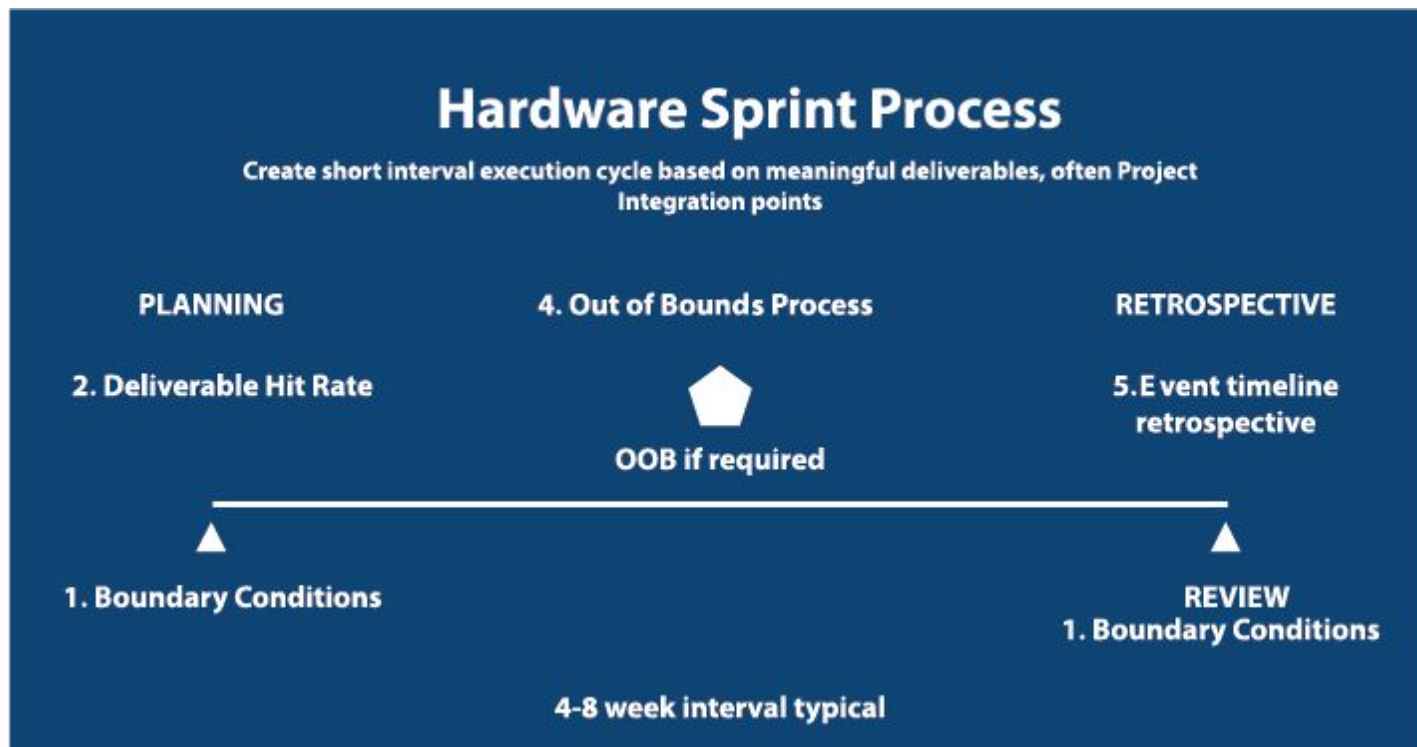
1. High performance teams
2. Self organized teams
3. Trust and empowerment
4. Customer owner & team interacting daily
5. Daily standup meetings

AND...

- Accept the fact that requirements may change

THEN, TRANSLATE SCRUM TO WATERFALL

1. User Stories into Boundary Conditions
2. Burn-down charts into Deliverable Hit Rate
3. Sprint to HW intervals
4. Manage the project with Out of Bounds Process
5. Sprint Retrospectives into Event Timeline Retrospectives



1. CREATING BOUNDARY CONDITIONS

- A program consists of product attributes and program attributes
 - Boundary conditions typically have both
- Create User Stories – Product Attributes
- Create budget and schedule – Program Attributes
- Select the top 3-7, define limits, and seek agreement with the management team



As a <type of user> I want <some goal> so that <some reason>

THIS BECOMES YOUR BOUNDARY CONDITIONS... STAY INSIDE THEM AND THE TEAM CAN KEEP MOVING FORWARD!

1. EXAMPLE BOUNDARY BREAK



Boundary Conditions

Agree on top 3-7 most important program and product requirements and set quantitative limits when possible.



- Deliverable Hit Rate too Slow!
- Key Engineer pulled!
- Three week delay!

2. TRANSLATE BURNDOWNS INTO DELIVERABLE HIT RATE

Identify the key requirements that should be satisfied during an interval

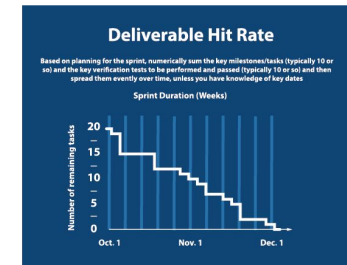
- Can be features implemented
- Can be tests validated
- Can be tasks performed
- Can be a customized metric of progress

This list of requirements can vary from interval to interval

- Front end is more definition loaded
- Middle is more task loaded
- Back end is more validation loaded

Create a target curve over the sprint interval

- Don't get too stressed out over perfection
- Assume that the events can be distributed evenly, unless you have clear knowledge otherwise

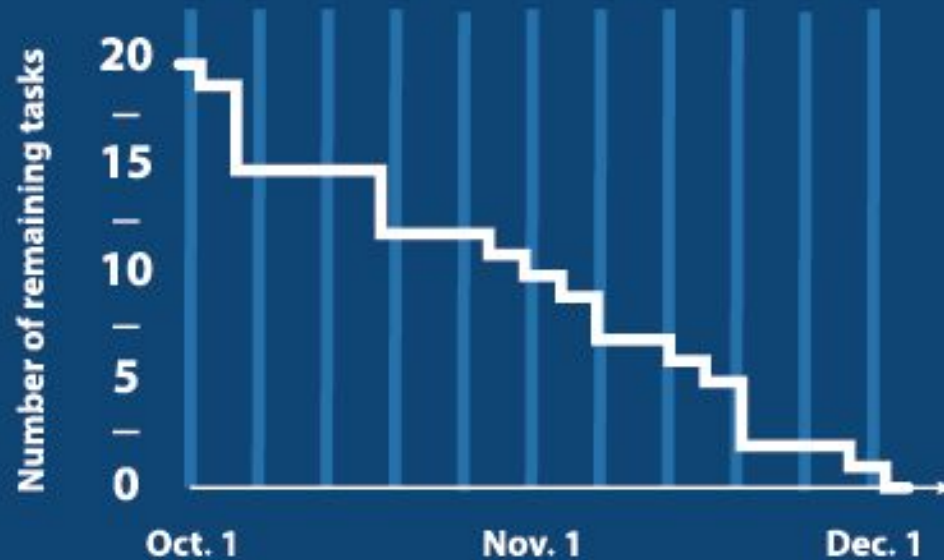


2. TRANSLATE BURNDOWNS INTO DELIVERABLE HITRATE

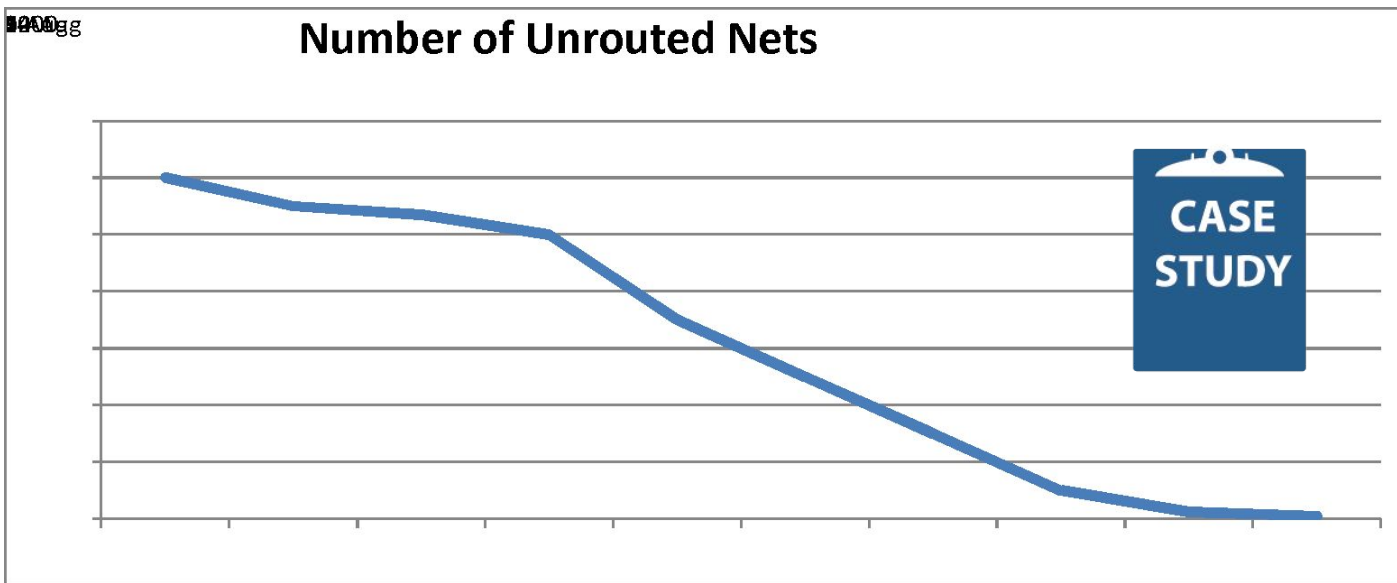
Deliverable Hit Rate

Based on planning for the sprint, numerically sum the key milestones/tasks (typically 10 or so) and the key verification tests to be performed and passed (typically 10 or so) and then spread them evenly over time, unless you have knowledge of key dates

Sprint Duration (Weeks)



2. EXAMPLE OF PCB LAYOUT PROGRESS



- Aug 1 started tracking PCB routing progress to get an idea of project velocity
- Aug 3 worried about progress, rate too slow
- Aug 4 increased # of engineers assigned to this task

Simple example of how the concept of a Burn Down Chart can be applied to see the team's progress in turning a schematic into a Printed Circuit Board layout

3. TRANSLATE SPRINT INTO HW INTERVALS

Continuous learning, short intervals, measurable progress, autonomy



**Idea
Approved**



**Concept
Approved**



**Lab Prototype
Accepted**



**Off Tool
Prototype
Accepted**



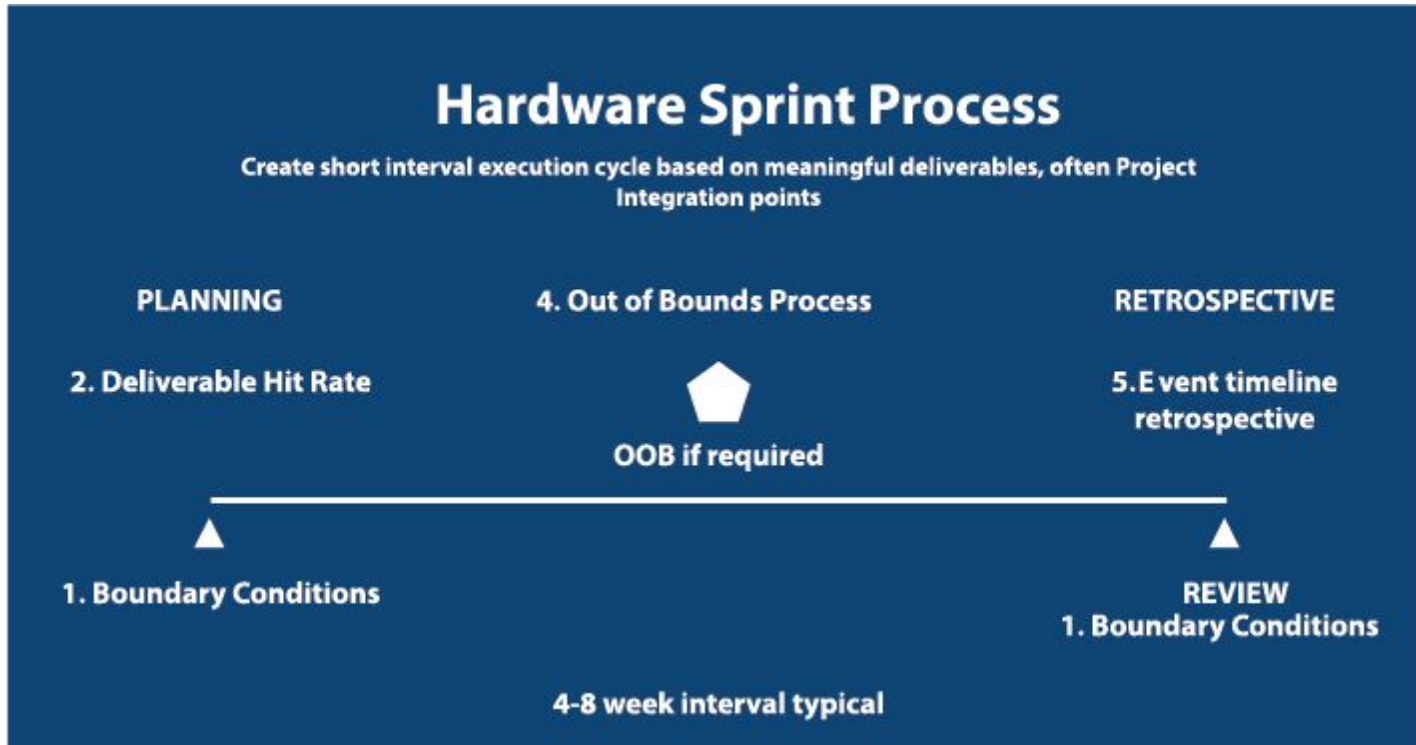
**Off Production
Process Accepted**



**Product
Released**

The true secret to getting the benefits of Agile development is to shed features/compromise completeness to not slip the interval

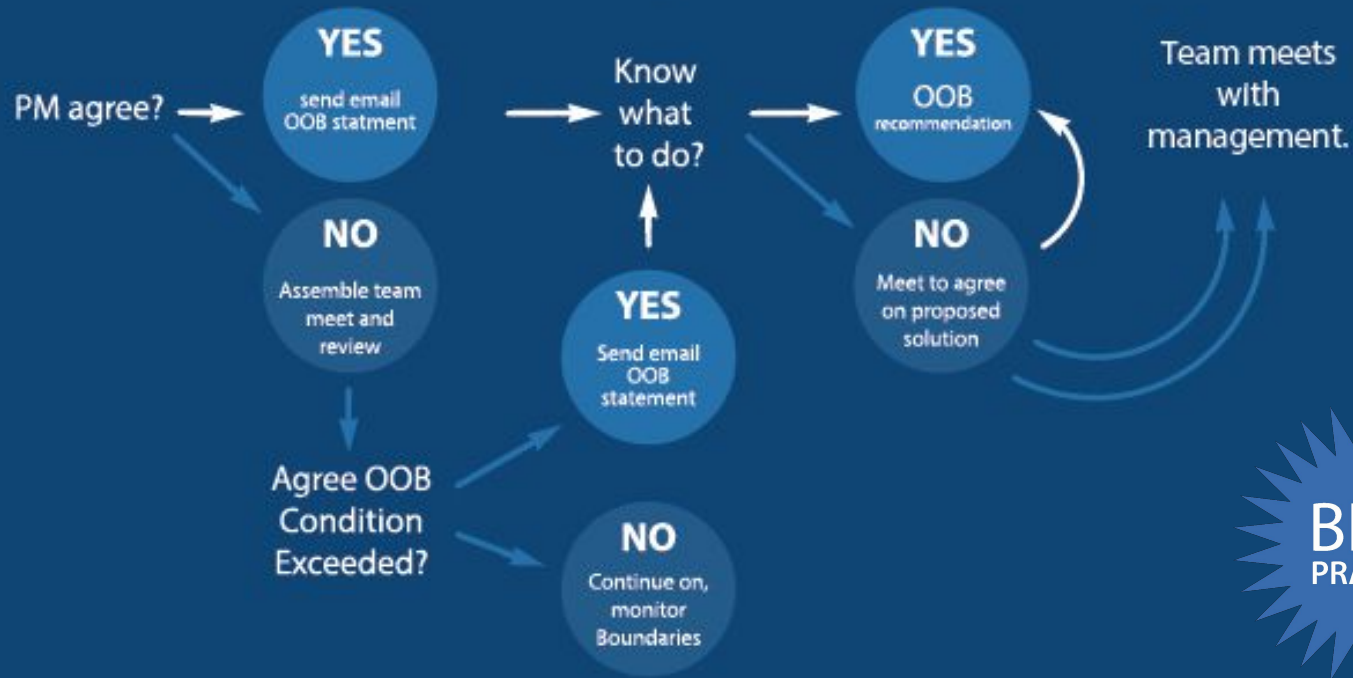
3. THE FIVE KEY ELEMENTS OF AGILITY



4. BOUNDARY CONDITION PROCESS

Out of Bounds Process

Description of the steps that the Project Manager follows when an OOB condition is known to be likely
This whole decision tree should take place in hours/days and not weeks/months



Event Time Line Process

Three Steps to a Productive Retrospective Review

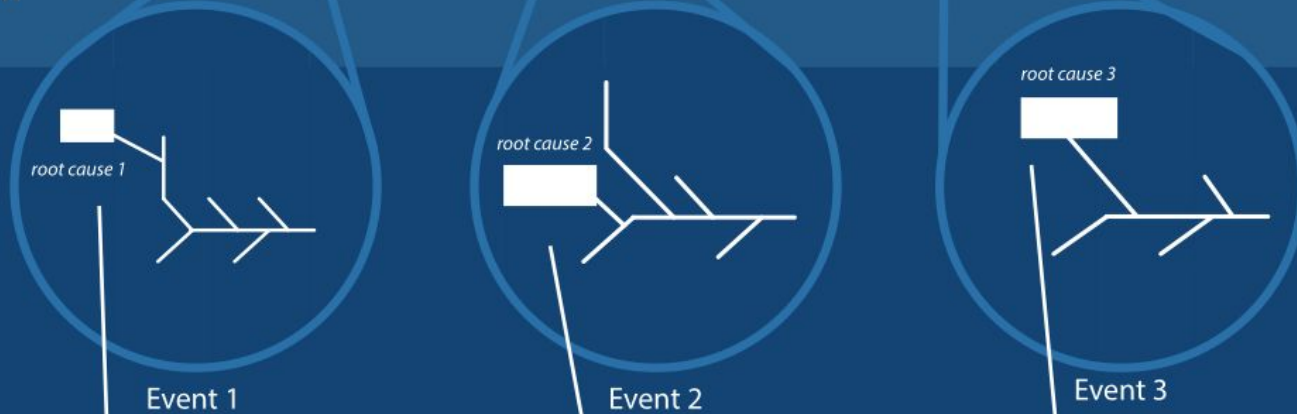
1. Event Analysis

Identify the impact of planned & unplanned events on a project outcome



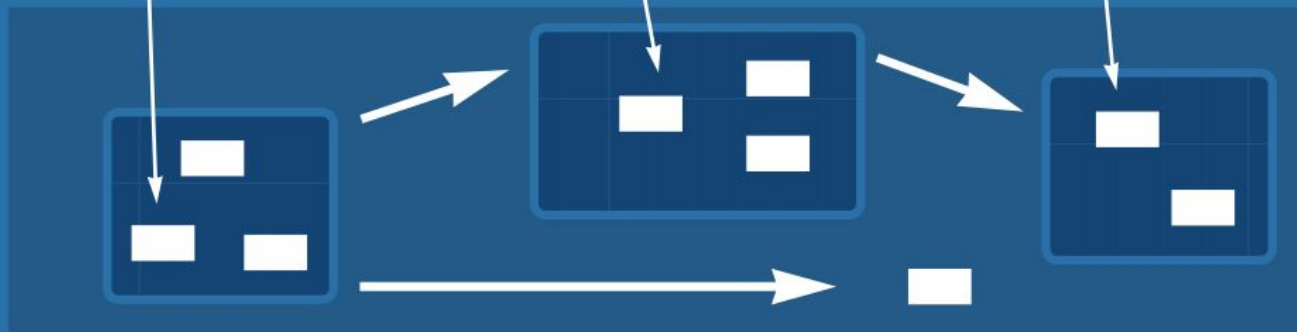
2. Root Cause Analysis

Select most significant root causes



3. Root Cause Synthesis

Understanding the big picture



Thank You!