Learning From The Trenches

SCRUM4HW
The Gathering

Jeanne Bradford
TCGen, Inc.
Cupertino, CA
August 2016

TCGen

Product Development & Strategy

- Architected Apple's product development process (ANPP) to gain scalability and speed
- Led development and program management teams for Apple, Cisco & Texas Instruments
- Certified Scrum Manager

TCGen

- Experience-based consulting firm practicing in product development and product strategy exclusively
- Focus on the critical few issues that are preventing clients from achieving their product development goals;

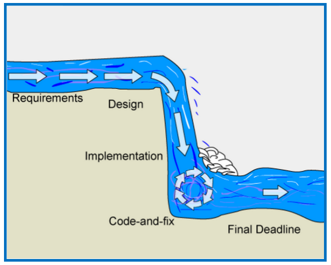
amazon cisco INTUITIVE SURGICAL® Roche

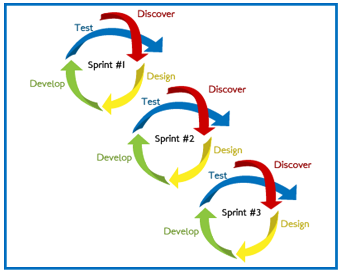
INNOVATE PRODUCTS FASTER
john carter | jeanne bradford
GRAPHICAL TOOLS FOR ACCELERATING PRODUCT DEVELOPMENT

Jeanne is co-author with John Carter of Innovate Products Faster a visual handbook on tools for teams.

Our Interest in Agile

- **Challenge**:
  - Our clients are delivering multi-platform products that integrate hardware, software, firmware, mobile components
  - Different PDP language; **integration is painful**; HW gets beat up for not being ”agile” (the verb)





- **Question**:
  - Can you make hardware integrations faster and less risky by using Agile methodologies?
- **Approach**:
  - Evaluating the Agile Manifesto
  - What can we learn from successful SW implementations & can that be applied beyond SW?

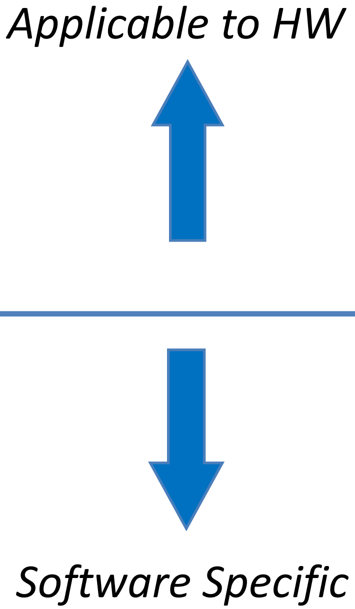
Agile Manifesto: Does it apply to Hardware?

75% of the Agile Manifesto CAN apply to development of any type

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

---

10. Continuous delivery of valuable software
11. Deliver working software frequently
12. Working software is the measure of progress



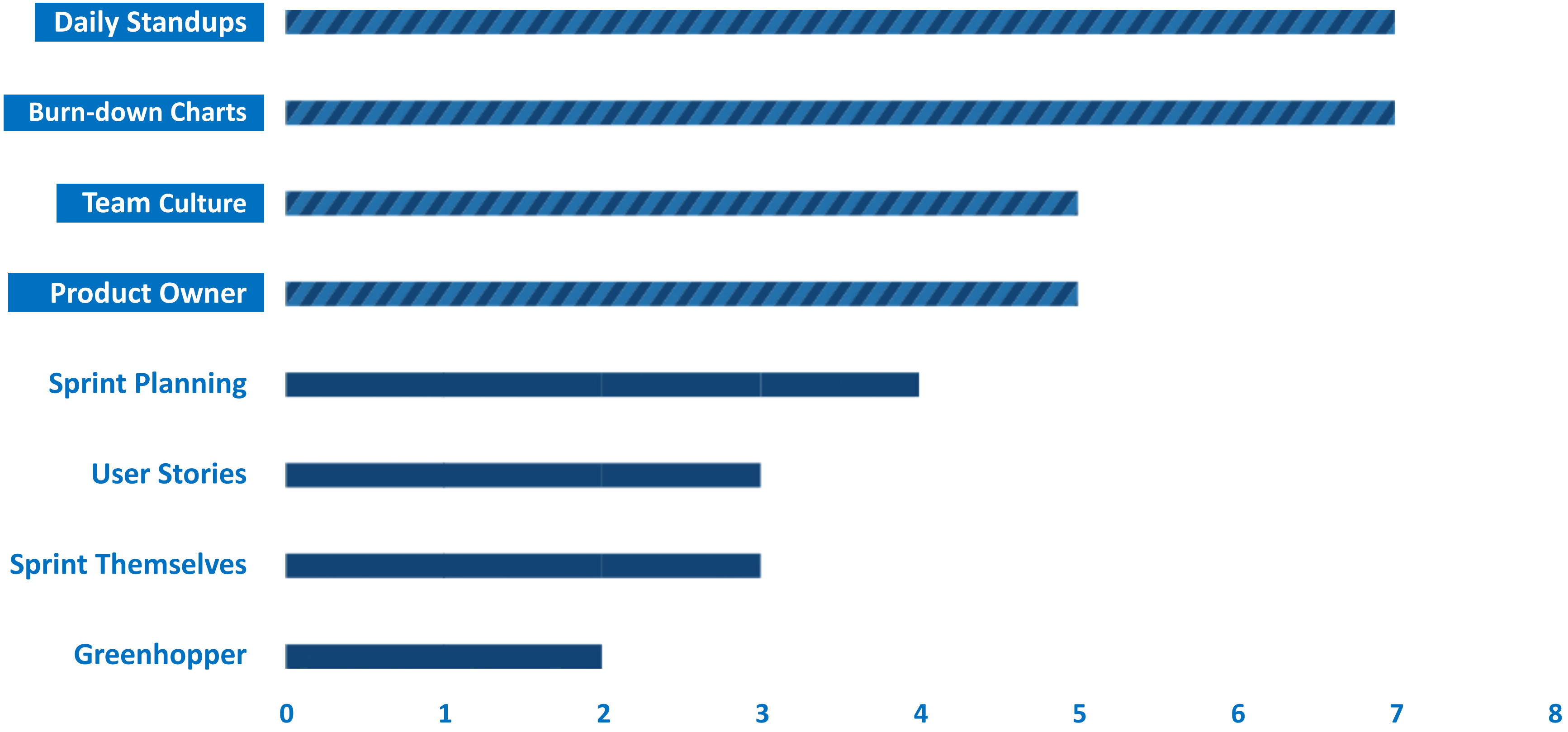
Applicable to HW

Software Specific

agilemanifesto.org

Agile Research – Most Impactful - SW

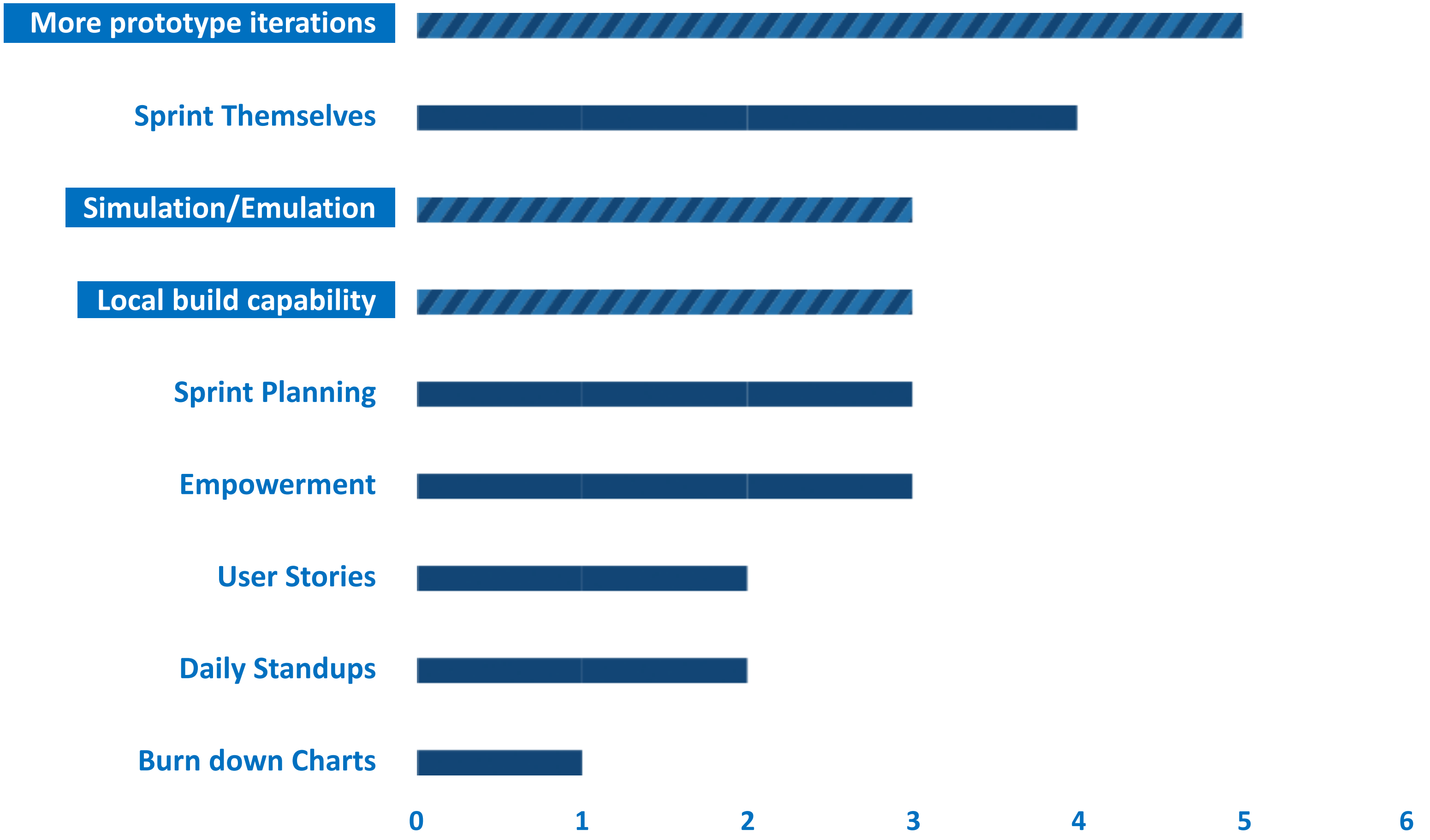
Question: What are the most impactful elements of Agile/Scrum applied to SW?



The top Scrum practices can be applied to Systems too!

Agile Research – Most Impactful – Hardware

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

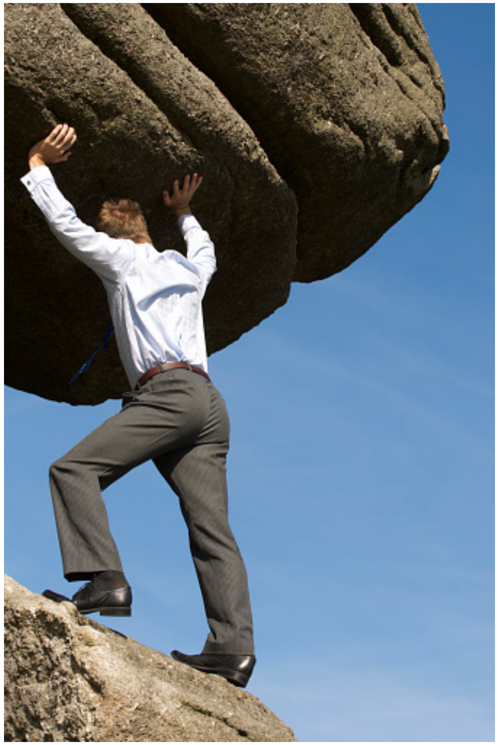


Three of the top "Agile" practices in Systems have little to do with Scrum

TCGen

Challenges of Applying Agile to Hardware

- Development partners/Supplier relationships
- Material Risk Buys:
  - Components with long lead times
  - Long supply chains with components, sub-assemblies, and final assemblies that need integration around the world
- Significant cost/time impact of late changes
- Medical products that require FDA compliance



There are significant differences that can't be ignored

Agile Methodology – Applied to HW/Systems

So can you apply Agile/Scrum to HW?

Agile Methodology – Applied to HW/Systems

Yes! But it's hard

And harder for hardware……

1. It requires a more sophisticated organization
   - ➢ Higher process literacy
   - ➢ Different decision-making model
   - ➢ Different roles & responsibilities
2. It's a new language in hardware
3. The tools **require *translation*** to be adaptable for hardware
4. Shorter intervals: no room to hide!

At the heart of applying Agile to hardware, tool translation & managing organizational change

Agile Tools & Skills

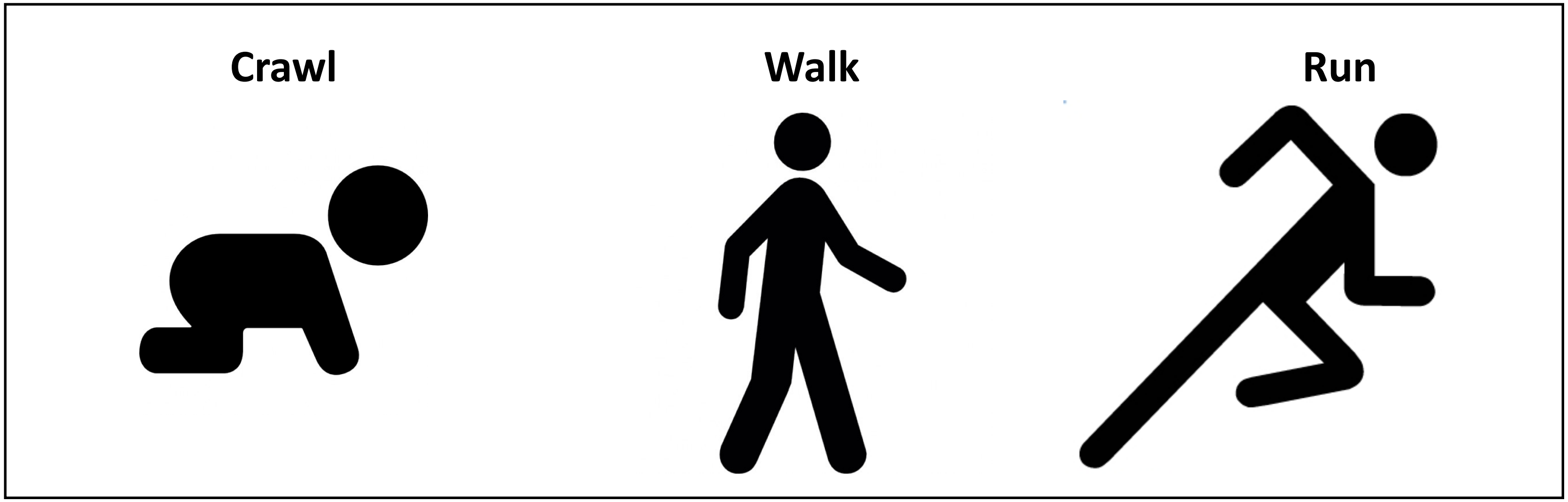
Translating Tools: Short Intervals

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

Organizational Change: High Performance Teams

1. Self organized teams
2. Trust and empowerment
3. Product owner & team interaction (near) daily
4. Risk management
5. Accountability

Approach to new processes



- People need a compelling reason for change
- People/teams change at different velocities
- HW people may not know anything about “Agile” methodologies
- Many organizational leaders think Agile is just for product teams

Organizational Challenges trumps Process/Tools!

Organizational Challenges Trumps Process/Tools

Agile Manifesto: self directed/high performance teams, but……

- Functional allegiance vs. product (customer) allegiance
- Redefines roles/responsibilities
- Management needs to trust/empower, teams need to be accountable

VS.

Organizational Challenges Trumps Process/Tools

(How coaches can add extreme value)

#1 Opportunity: Today ➔ Tomorrow

- Teams cannot go straight from the training room to their desks, and start applying Agile/Scrum
- Challenge: to break the momentum of what they were doing yesterday in order to behave differently today

# 2 Opportunity: Functional ➔ Product Allegiance

- If teams are self directed and empowered, what value do middle managers provide?
- Requires behavior shift: servant leadership
- Functional leaders/peers need to have a collaboration mindset

If you don’t take on the organizational piece, the tools/processes will be minimized

How to take on these opportunities

1. Break the momentum of what they were doing yesterday in order to behave differently today
   - ”As-Is” → “To-Be”
     - Clarify new roles and responsibilities
     - Identify supporters and detractors (Attitude-Influence)
   - How to move from waterfall to Agile
     - Crawl, walk, run

2. Moving from Functional to Product Allegiance
   - Collaboration up and down the organization
   - IQ vs EQ

WHAT? SORRY. I WAS USING THIS TIME TO THINK ABOUT SOMETHING USEFUL.
Dilbert.com DilbertCartoonist@gmail.co
1-27-10 ©2010 Scott Adams, Inc./Dist. by UFS, I

Managing Change By Measuring Behavior

Implementing new processes requires a behavior change

- What is the earliest moment that you can know whether or not teams are executing to the new process?
- ***Predictive metrics*** inform you at the earliest moment that will tell you if people are changing their behavior
  - Simple, easy to measure, frequently measured

Example: Your team need to hire 5 engineers by the end of the quarter. What metric do you show in a management review?

Managing Change By Measuring Behavior

Implementing new processes requires a behavior change

- What is the earliest moment that you can know whether or not teams are executing to the new process?
- ***Predictive metrics*** inform you at the earliest moment that will tell you if people are changing their behavior
  - Simple, easy to measure, frequently measured

Example: Your team need to hire 5 engineers by the end of the quarter. What metric do you show in a management review?

The best predictive metric?

Managing Change By Measuring Behavior

Implementing new processes requires a behavior change

- What is the earliest moment that you can know whether or not teams are executing to the new process?
- ***Predictive metrics*** inform you at the earliest moment that will tell you if people are changing their behavior
  - Simple, easy to measure, frequently measured

Example: Your team need to hire 5 engineers by the end of the quarter. What metric do you show in a management review?

The best predictive metric?

Number of screening calls per day by hiring manager

As coaches, could applying Predictive Metrics help your success?

Work Session

Use your experience (or desired experience) with leading change

- What would a successful out come be?
- What specific behavior would be changed, and who’s behavior is that?

Assignment:

1. Describe the process change: As-Is ➔ To Be
2. Identify 3-4 required behavioral changes
3. Choose one, and describe:
    - What’s the behavior
    - How would you measure it
    - How often would you measure it
    - How would the data inform you? (action)

Thank You!

Contact:

Jeanne Bradford

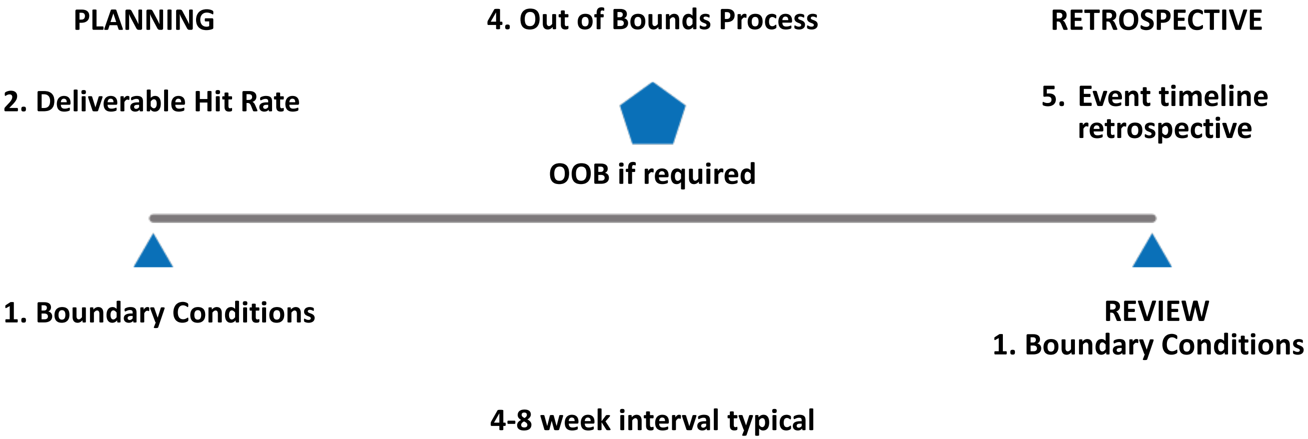
jbradford@tcgen.com

Translating Scrum Tools to Hardware

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

Hardware Sprint Process

Create short interval execution cycle based on meaningful deliverables, often Project Integration points



PLANNING

2. Deliverable Hit Rate

1. Boundary Conditions

4. Out of Bounds Process

OOB if required

RETROSPECTIVE

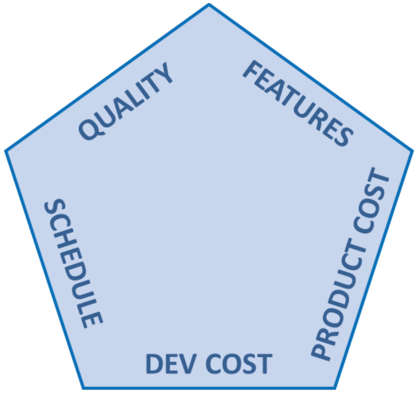
5. Event timeline retrospective

REVIEW
1. Boundary Conditions

4-8 week interval typical

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!

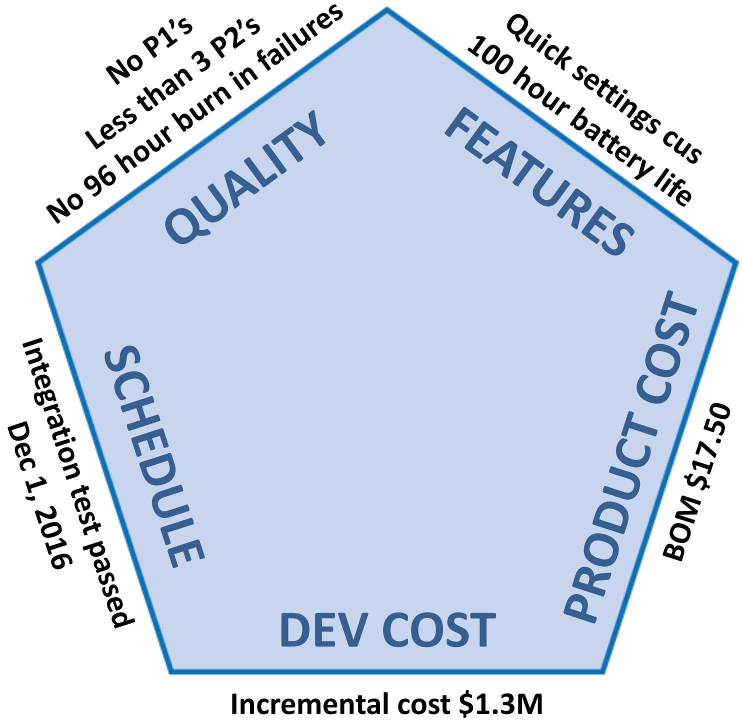
www.mountaingoatsoftware.com/topics/user-stories

1. Example Boundary Conditions

Boundary Conditions

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

BEST PRACTICE



QUALITY

No P1's
Less than 3 P2's
No 96 hour burn in failures

FEATURES

Quick settings cus
100 hour battery life

PRODUCT COST

BOM $17.50

DEV COST

Incremental cost $1.3M

SCHEDULE

Integration test passed
Dec 1, 2016

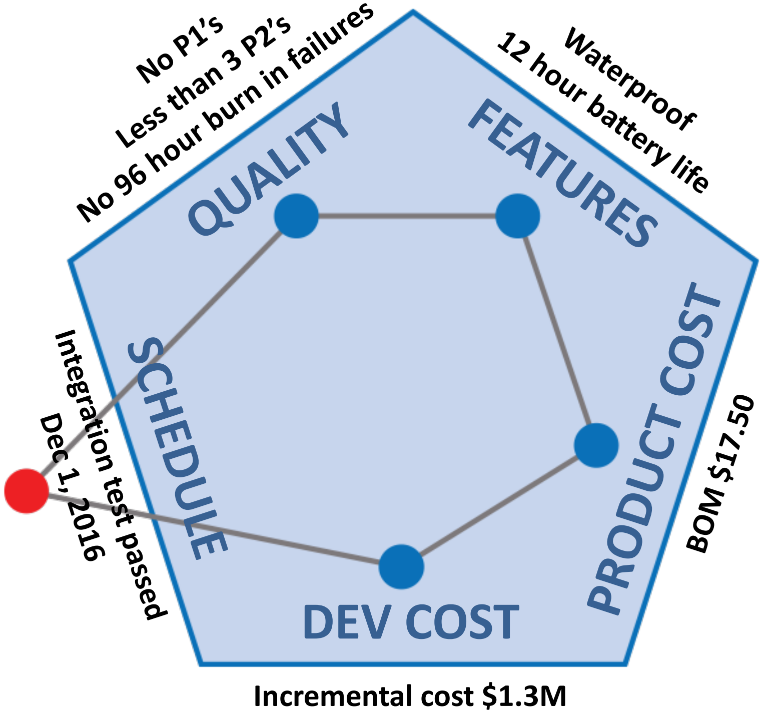
1. Example Boundary Break

Boundary Conditions

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

BEST PRACTICE

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



TCGen

2. Translate Burn downs Into Deliverable Hit Rate

Identify the key tasks that should be satisfied during an interval

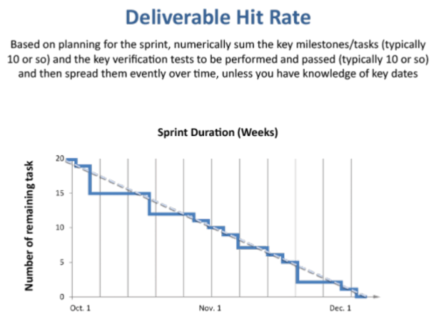
- Can features be implemented?
- Can features/specs be validated?
- Can tasks be performed?
- Can be a customized metric of progress

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



Deliverable Hit Rate

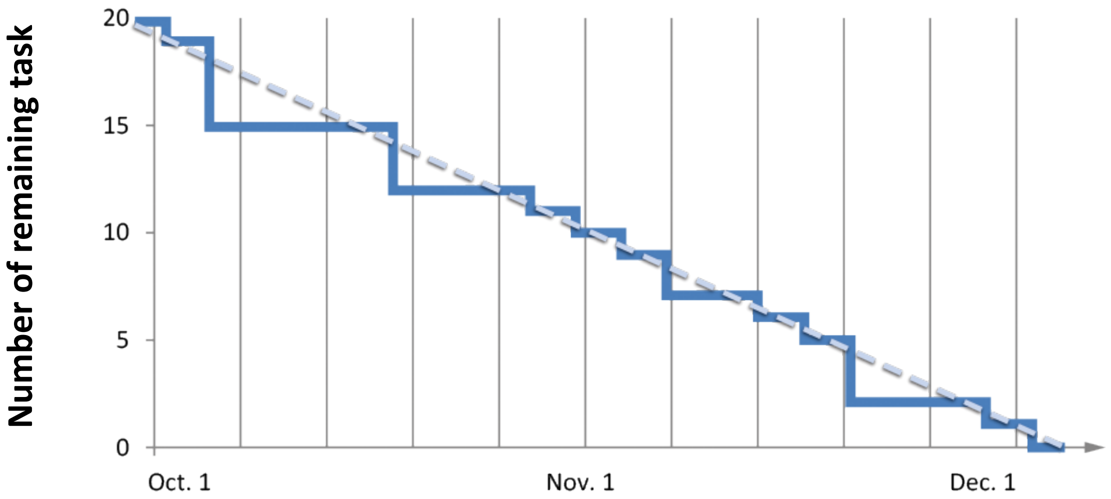
Based on planning for the sprint, numerically sum the key milestones/tasks (typically 10 or so) and the key verification tests to be performed and passed (typically 10 or so) and then spread them evently over time, unless you have knowledge of key dates

2. Translate Burn downs Into Deliverable Hit Rate

Deliverable Hit Rate

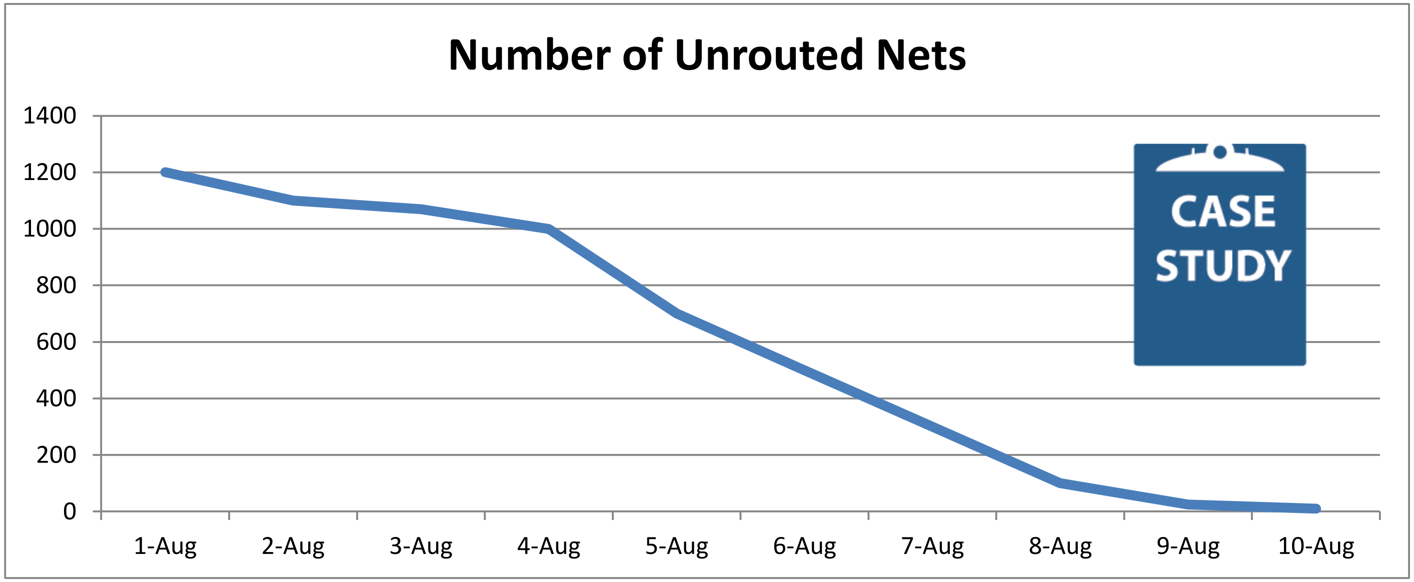
Based on planning for the sprint, numerically sum the key milestones/tasks (typically 10 or so) and the key verification tests to be performed and passed (typically 10 or so) and then spread them evenly over time, unless you have knowledge of key dates

Sprint Duration (Weeks)



What's the right duration for a HW Sprint?

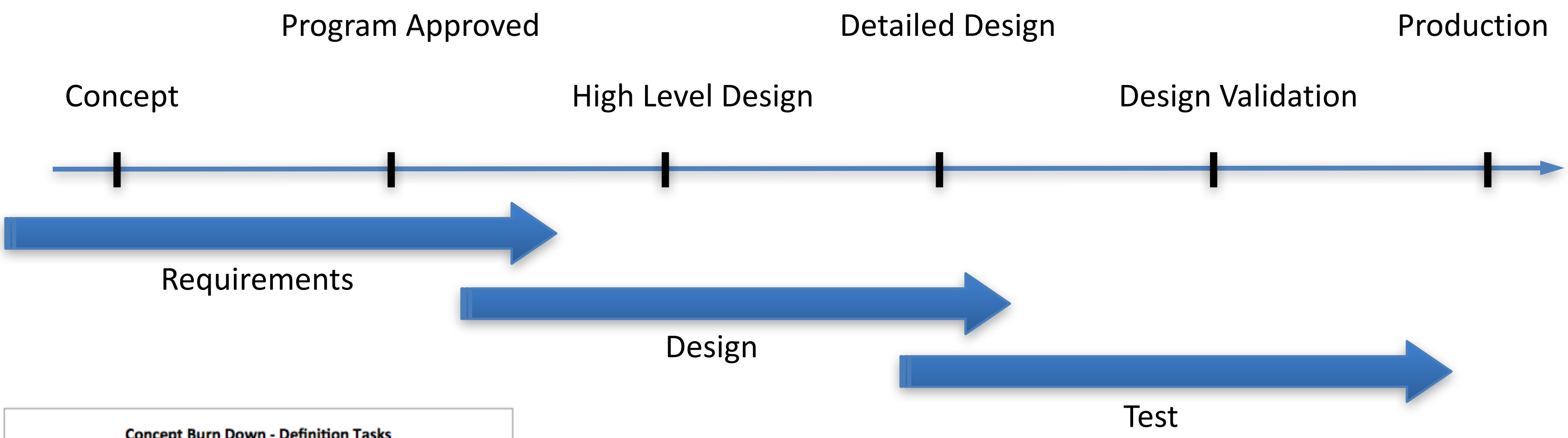
2. Example of PCB Layout Progress



- 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

Example how a Burn Down Chart can be applied to see the progress in turning a schematic into a layout

Burn Down Metrics for Hybrid Development


Program Approved

Detailed Design

Production

Concept

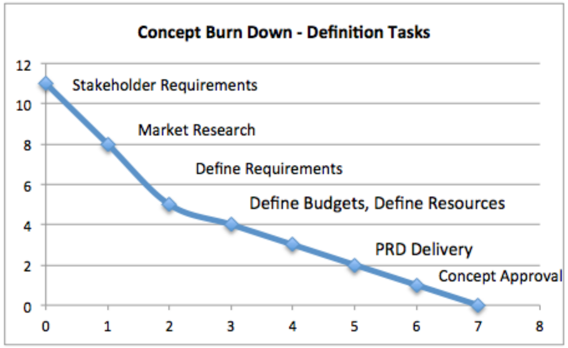
High Level Design

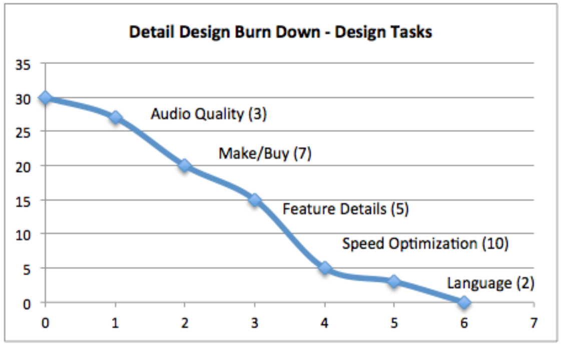
Design Validation

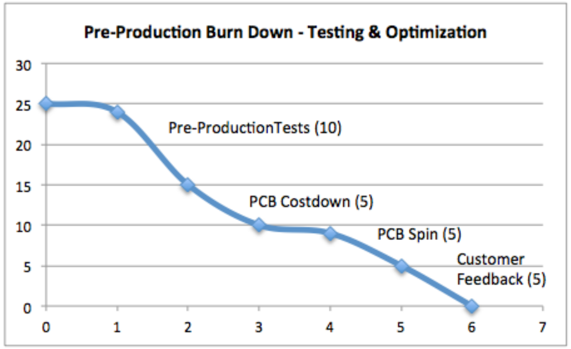
Requirements

Design

Test


Concept Burn Down - Definition Tasks

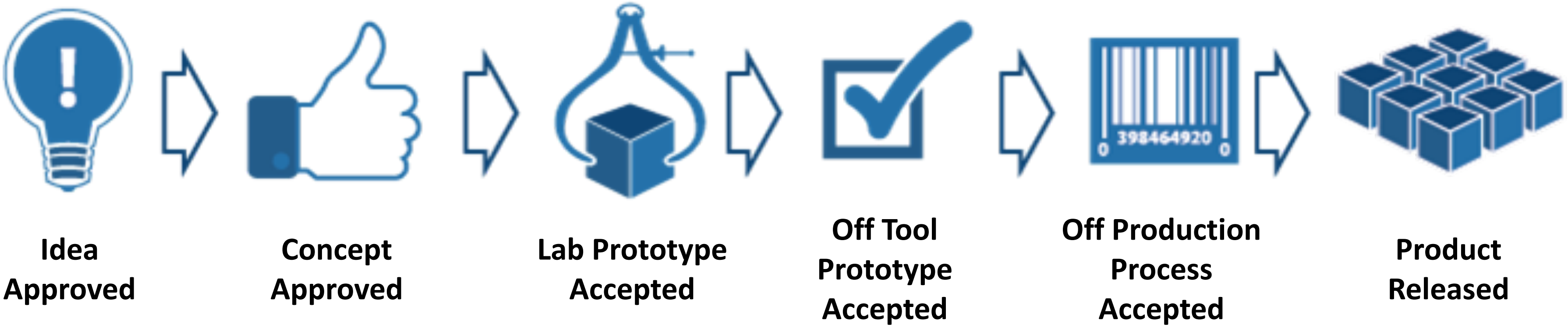

Detail Design Burn Down - Design Tasks


Pre-Production Burn Down - Testing & Optimization

3. Translate Sprint Into HW Intervals

Divide the project into the smallest increment possible that represents TRUE INTEGRATION POINTS or CLEARLY DEFINABLE MILESTONES.

BEST PRACTICE



Idea Approved

Concept Approved

Lab Prototype Accepted

Off Tool Prototype Accepted

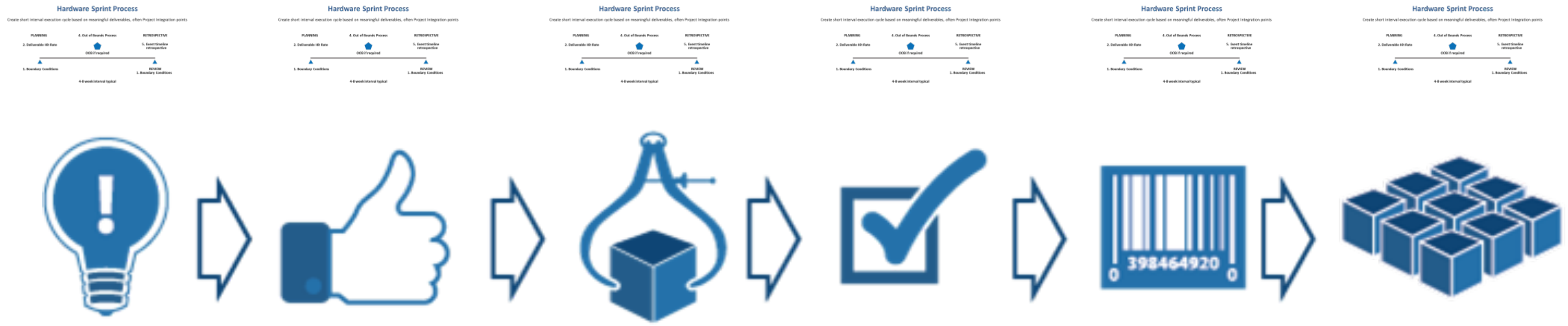
Off Production Process Accepted

Product Released

CASE STUDY

3. Translate Sprint Into HW Intervals

Continuous learning, short intervals, measurable progress, autonomy



Idea Approved

Concept Approved

Lab Prototype Accepted

Off Tool Prototype Accepted

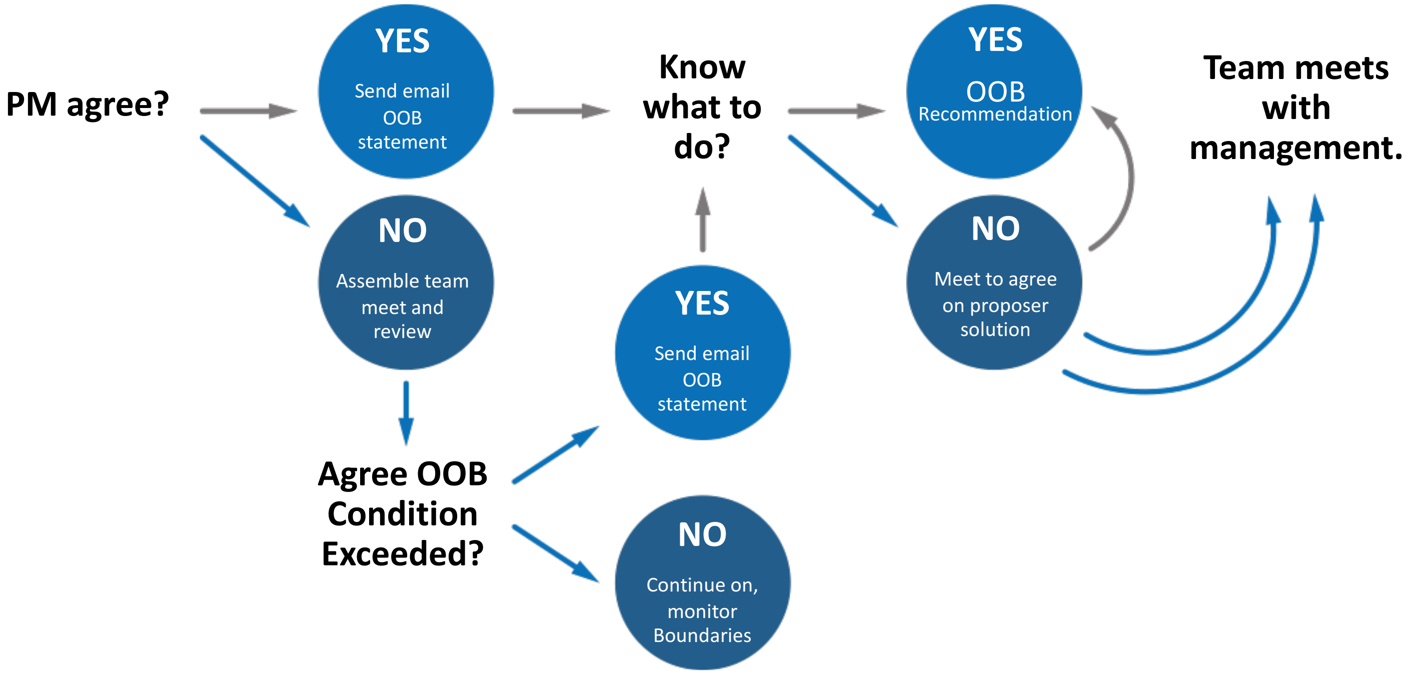
Off Production Process Accepted

Product Released

The secret to getting the benefits of Agile development is to not slip the interval

Out of Bounds Conditions

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.


PM agree?
YES
Send email OOB statement
NO
Assemble team meet and review
Agree OOB Condition Exceeded?
YES
Send email OOB statement
NO
Continue on, monitor Boundaries
Know what to do?
YES
OOB Recommendation
NO
Meet to agree on proposer solution
Team meets with management.

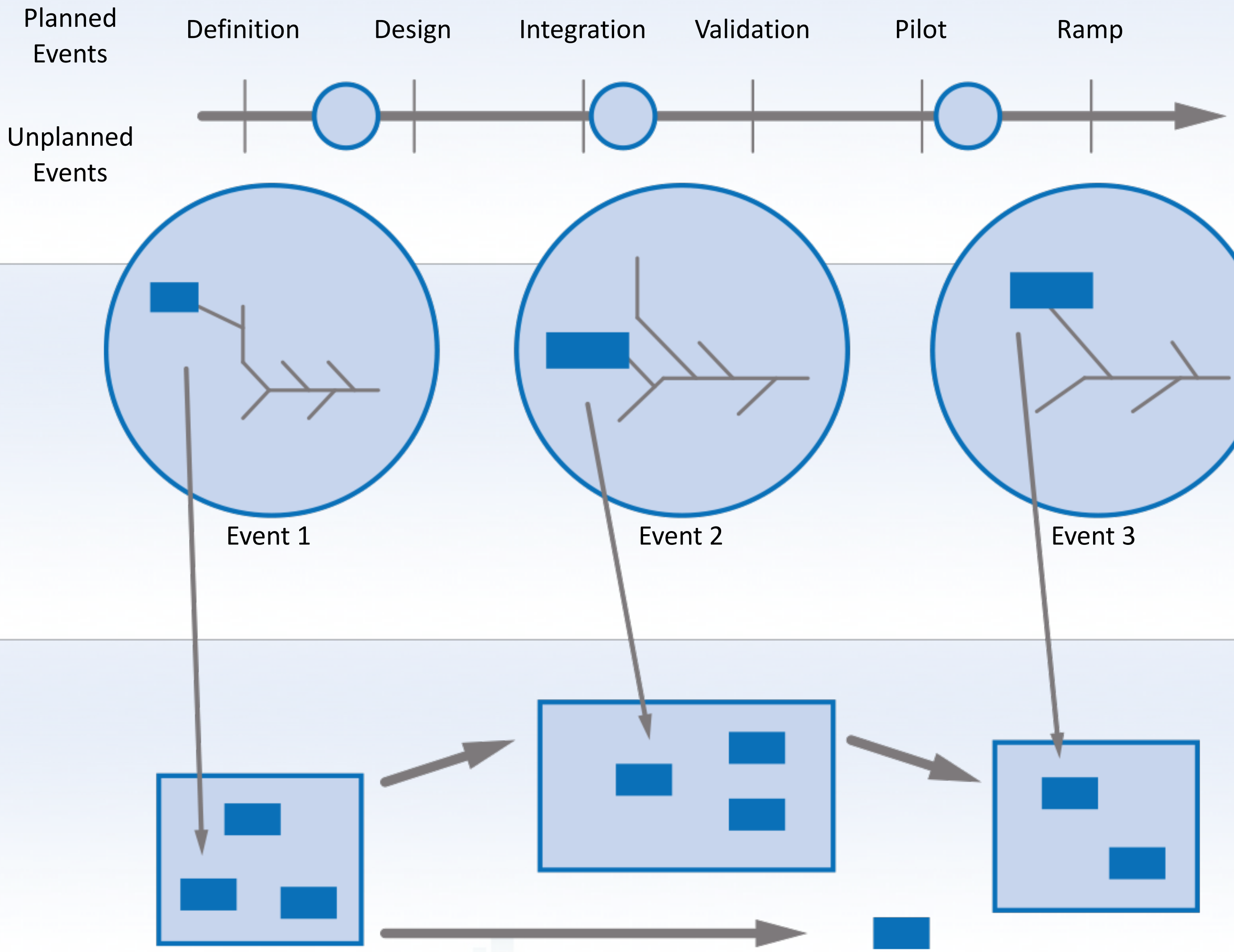
5. Event Timelines and Retrospectives

Event Timeline Process
Three Steps to a Productive Retrospective Review

1. **Event Analysis**
   Identify the impact of planned & unplanned events on project outcome

2. **Root Cause Analysis**
   Select most significant root causes

3. **Root Cause Synthesis**
   Understanding the big picture



Retrospectives

- Retrospectives should be carried out on all programs
- The retrospectives should follow a common process which has the following attributes
  1. Fact based, and data driven
  2. Involve Cross-functional team members
- The retrospective process should be owned by the team
- The retrospective process should be used during every Interval
- The process consists of the following steps
  1. Event time lines & Prioritization of the biggest events
  2. Root Cause Analysis
  3. Affinity Diagram to summarize results