Agile Beyond Software
Applying Agile Methods to Tangible Products

IEEE Technical Engineering Management Society

March 2017

John Carter
Jeanne Bradford
TCGen, Inc. Menlo Park, CA

TCGen

Agenda

- Agile for tangible products?
  - Our path to the realization that…
- Key insights: What we've learned/what surprised us
  - Summaries of research we have performed
- Short Intervals & High Performance Teams
- Five components of PRACTICAL Agile
  - Foundation principles
  - Nested Sprints
- Breakout session: Planning a sprint
  - Sprint Planning 'One Pager'
  - Sprint Score: Grading your effort!

• • •

TCGen: Approach & Background

Approach: **Allegiance to product development goals, not a specific process**

1. Focus on unmet product development goals
2. Uncovering & aligning on most impactful root causes
3. Customized solutions to fit culture & current capability
4. "Inch-wide, mile-deep" sustainable best practices
5. Managing Change by Measuring Behavior

3M, Abbott, Agilent Technologies, Apple, Baxter, BD, bmc, BOSE, Cisco, Dolby, fitbit, Helen of Troy, hp, IBM, Intuitive Surgical, Medtronic, Roche, WD, Xerox

John Carter

- Board of Directors of Cirrus Logic (CRUS)
- Raised private equity; executed rollup as a Principal & CTO of Klipsch Group
- Chief Engineer of BOSE; holds original patent on Noise Cancelling Headphones

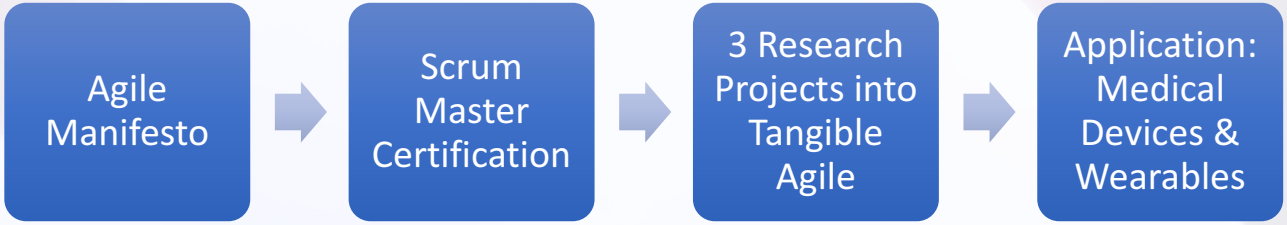
Jeanne Bradford

- Architected Apple's product development process (ANPP) to gain scalability and speed
- Led development and program management teams for Apple, Cisco & Texas Instruments
- Builds high performance teams to accelerate product development

Our Path To Agile Solutions Beyond Software

In our work with technology clients, we notice that an organization-wide shift is taking place with companies that deliver a **multi-platform solutions** (HW, SW, FW, Mobile and/or Web):

- Product delivery is accomplished by multiple development teams; typically in **silos** until they are forced to integrate
- The **integration points were not well planned**, due to the silos
- **As software teams** adopt Agile methodologies, they deliver **faster,** are more **adaptive**, and have a better understanding of their **customer**
- There are now **multiple** product development processes with different **language** and **roles**. This further aggravated integration.
- There is **increasing pressure** on hardware teams to become **"more Agile"**, like their SW & FW colleagues



Agile Manifesto – Just for Software?

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

Software Specific

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

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

agilemanifesto.org

Agile Success Stories & Challenges

No shortage of Agile success stories in the SW world

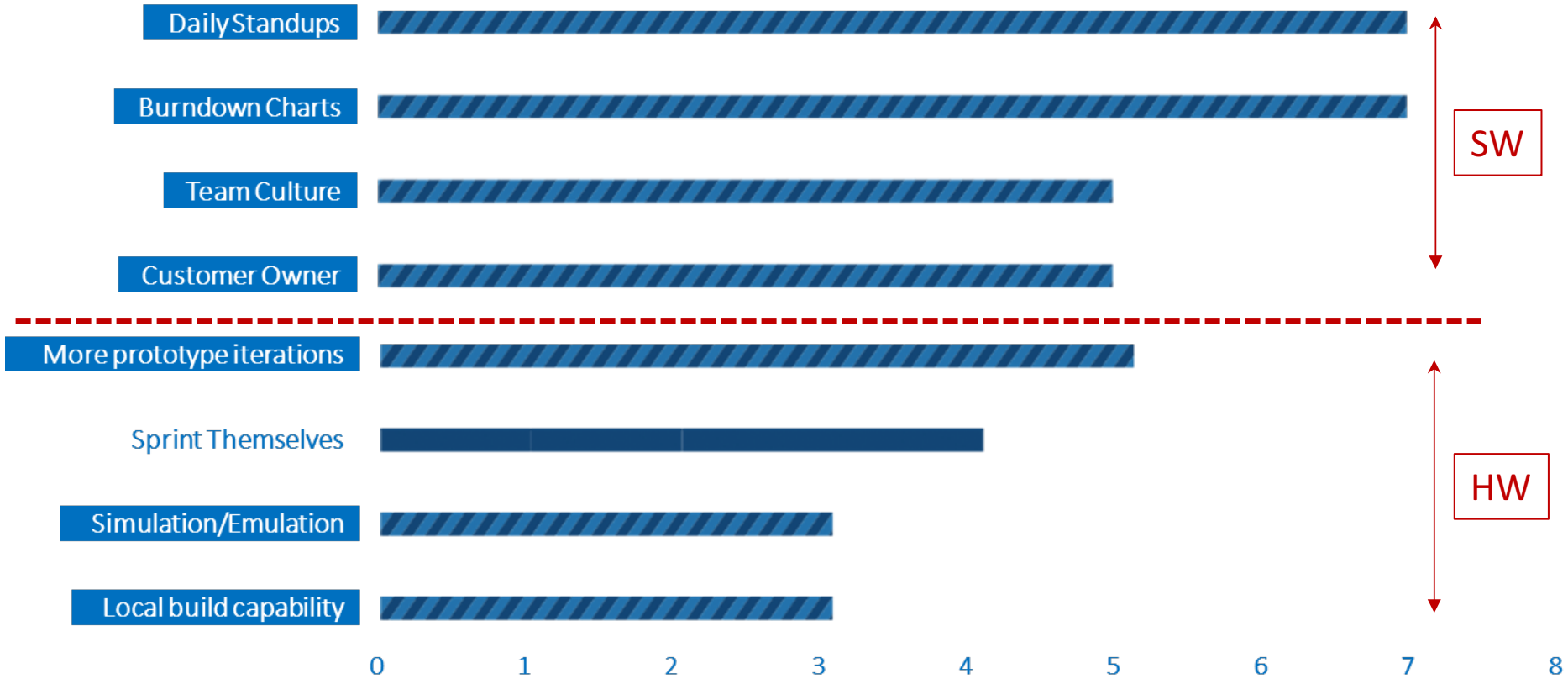
But all of these successes were in SW… how can we apply to Physical Products/Systems?

- Component lead times
- Materials risk buys
- Suppliers with their own processes

Conducted research of case studies where Agile methods were successfully applied with nearly 20 participants

Agile Methodology – Best Practices

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



Top Agile practices can be applied to tangible products:
*Focus on **high performance teams** & **short intervals***

Guiding Principles

- No roadmap for doing Agile for Hardware
- Most teams outside of software have **no knowledge of Agile** (tools, terminology) – And don't believe that it applies
- Tools need to be translated to work for tangible products
- A Sprints/Scrum model was the best Agile tool to adapt/apply
- Hardware sprints are best with **longer durations**, and might vary in length between phases

Selecting and adapting the best Agile methods can lead to faster delivery

What We Learned

- Team **composition is more complex than software teams**: mechanical engineering, supply chain, industrial design
  - *The work isn't different, but how you organize and execute can be improved*
- **User stories** are not as effective with tangibles:
  - *Sprints defined by **functional** stories and **closing knowledge gaps***
- Sprints within a Waterfall framework accelerate decisions, retire risk faster
  - Multiple sprints within a milestone create a more organized sense of urgency; and less chaos
  - But….there's no room to hide
- Teams were skeptical, but started seeing gains after the first few sprints

Teams focused on learning faster, got the core value of sprints

What Surprised Us

- Breaking away from a waterfall process is **too big of a first bite** for most hardware teams
  - *Fundamentally an organization change management issue.*
  - *People change at different velocities, so don't clear the deck, apply solutions surgically.*
- Nesting sprints within an existing Waterfall process is better than Agile/sprints alone
  - *The focus is not on the next milestone (for tangible products months away), but on the next 2-3 weeks.*
  - *Nested sprints within milestones, focus on rapid prototyping and key decisions (knowledge gaps)*
- Most team implementing Waterfall are ignoring fundamental best practices
  - *A PERT can get you from gate to gate*
- Organizations desire to **"be more agile"** not "to do Agile"

Integrating Agile Sprints within a Waterfall framework can be the best of both worlds

Agile for Tangible Products – Puts and Takes

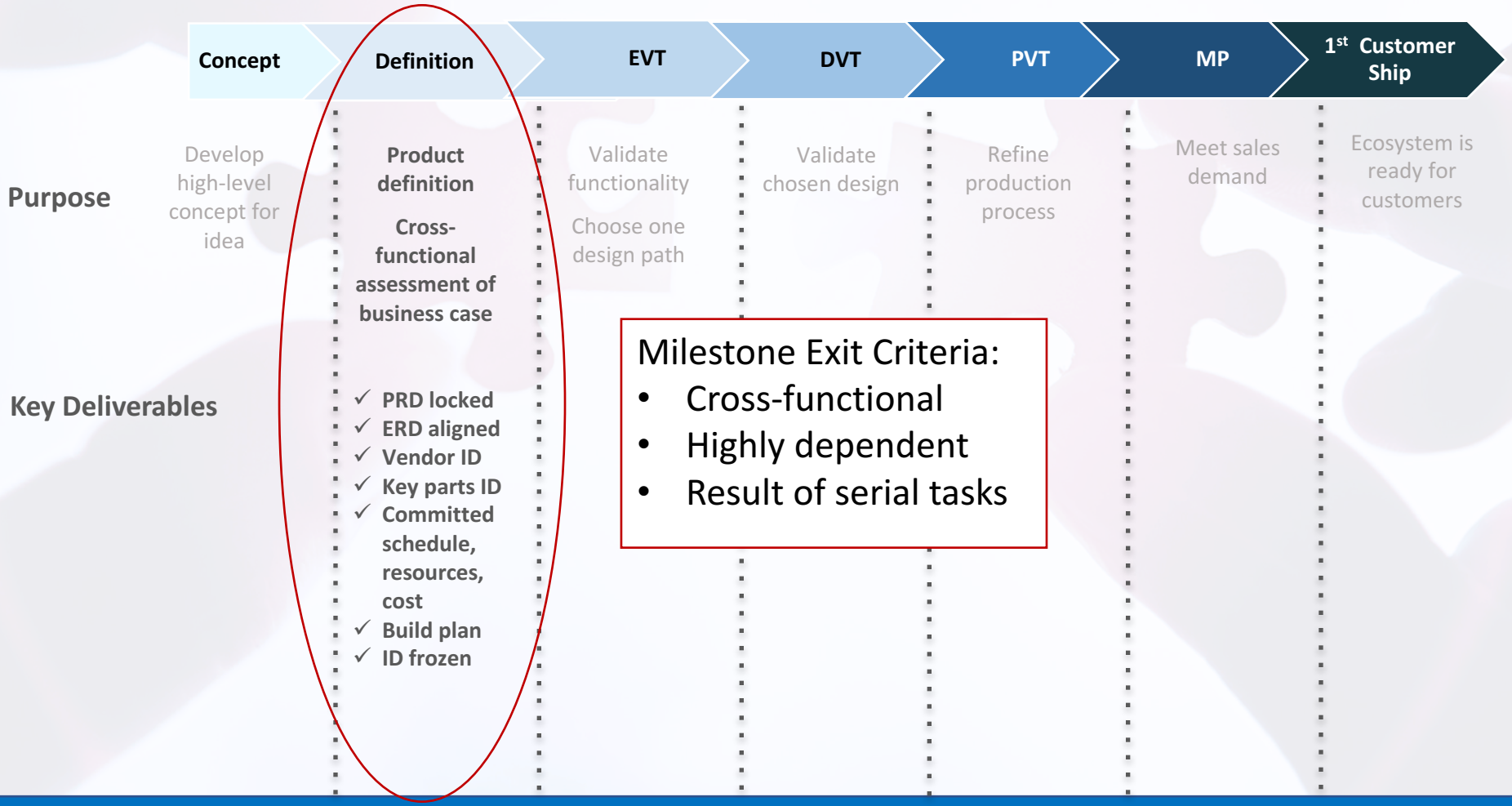
	Benefit
Increase speed of development	+++
Reduce team burnout	+++
Closer to the customer	++
Accommodates changes in Requirements	++
Performs testing earlier	+
Easy, uniform tracking (Story Points)	-
Sprints identical – easier to implement	--
Provides capability to ship MVP	---
Ships more usable code	---

1.5

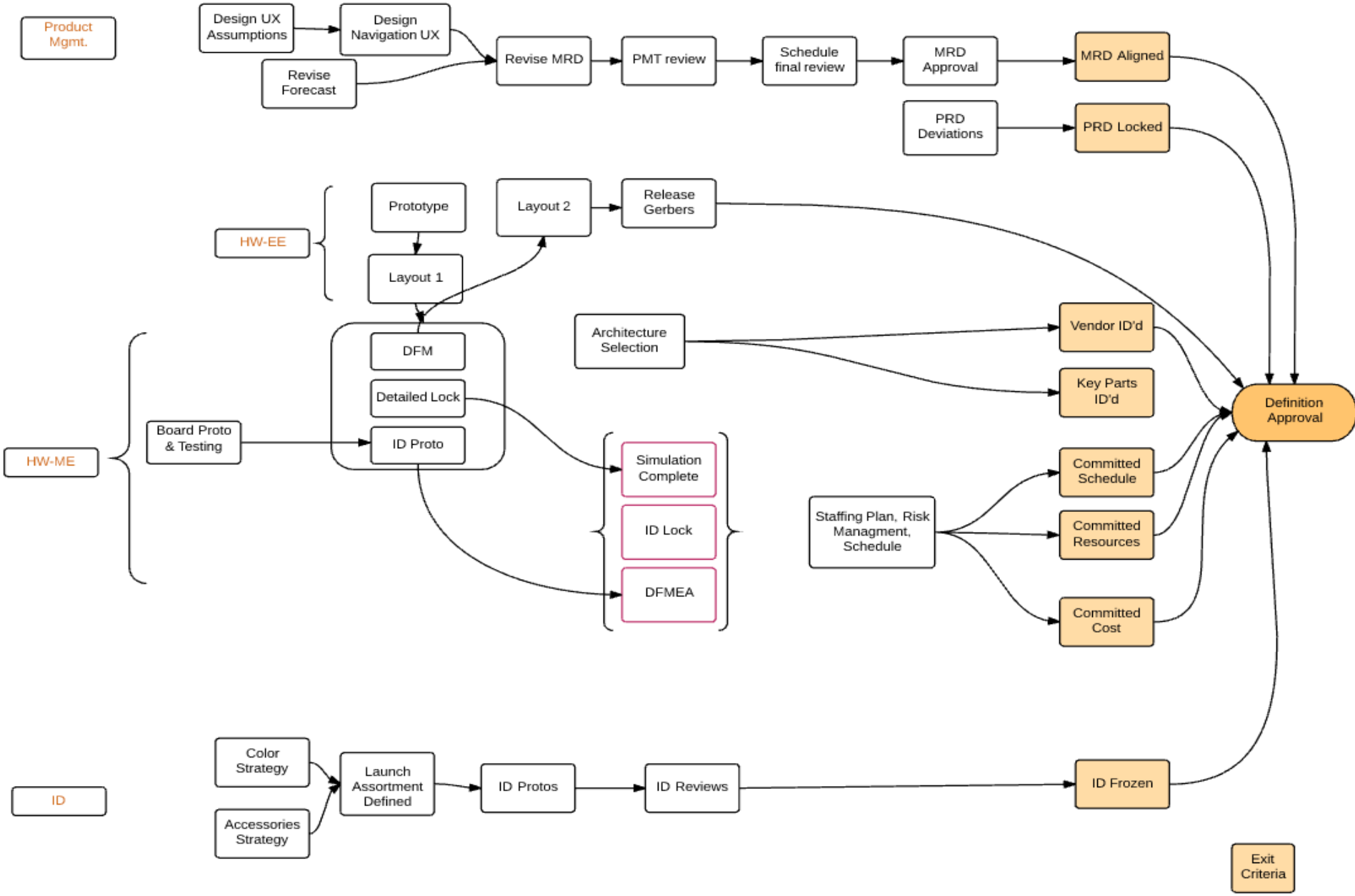
There are some areas where Agile for Software is fundamentally better
But that does not mean that substantial benefits can be gained

Short Intervals : Nest Sprints within Milestones

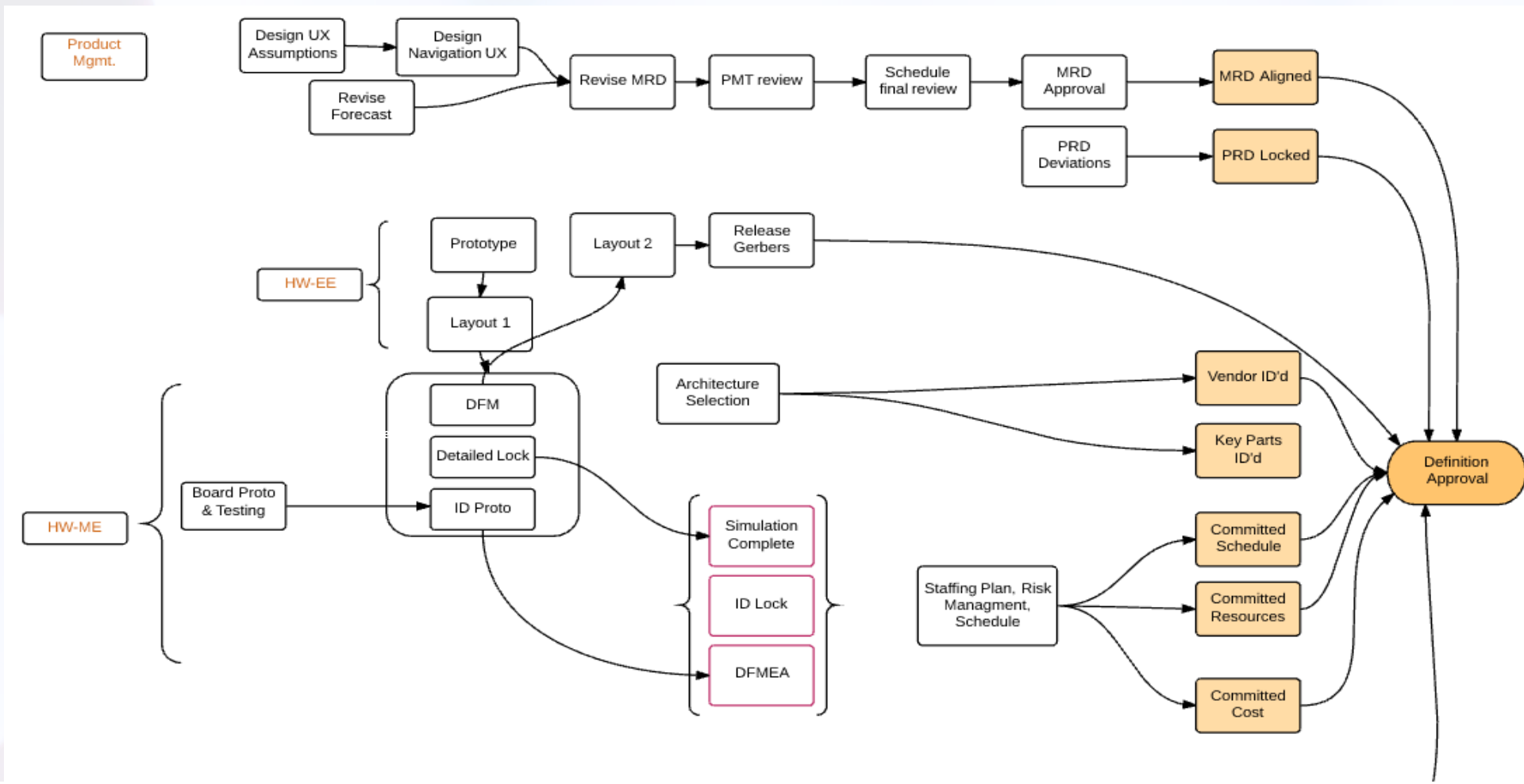
- We're not suggesting you throw waterfall out – quite the contrary.
- Short intervals can optimize milestone achievement



Short Intervals : Blending Waterfall & Agile



Short Intervals : Blending Waterfall & Agile



WATERFALL MILESTONE N

WATERFALL MILESTONE N+1

NEW - NESTED SPRINTS

P Execute/Measure R P Execute/Measure R P Execute/Measure R P Execute/Measure R P Execute/Measure R

EXISTING - PHASE OF CURRENT PROCESS

High Performance Teams: *Teams over Functions*

1. Shared vision of success
2. Self organized teams
3. Trust and empowerment
4. Team interacting daily
5. Work through tough times together

Biggest challenges:

- Agile require a stronger team skills than Waterfall.
- The impact of functional allegiance over product allegiance is amplified
- Teams are “empowered”, but don’t have sufficient domain or collaboration skills
- People accept change at different velocities -> inconsistent capabilities

Time is gained or lost every day at the team level

High Performance Teams: *Teams over Functions*

1. Shared vision of success
   - *Project Kick-offs? Or just 'get started'?*
2. Self organized teams
   - Teams wait to be told what to do?
   - Teams understand project goals, exercise team allegiance over functional allegiance
3. Trust and empowerment
   - Management meddles; reviews, reports, micro-manages?
   - Review burndown charts; use boundary conditions
4. Teams interact often
   - Are weekly team meetings really effective?
   - Sprints will require more frequent and more productive meetings.

When Bad Things Happen….?

OR

You won't optimize product delivery, until you master HPTs

Five Key Components for Successful Implementation

Five Elements of Agile Development

1) High Performance Teamwork: Teams over Functions

2) Sprint Planning

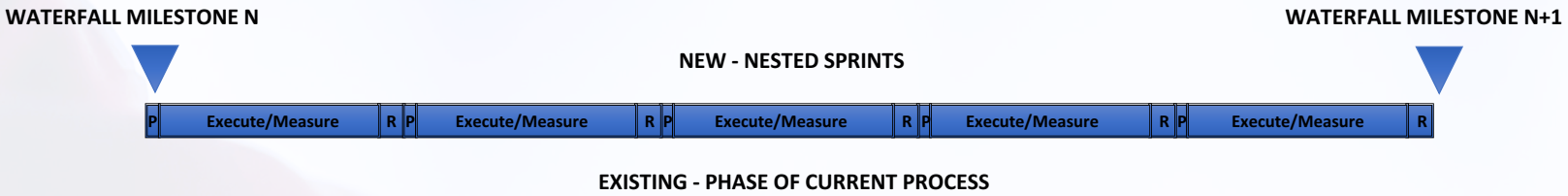
3) Executing/Measuring

4) Reviewing

5) Nesting Sprints in Waterfall Framework

Waterfall – Keep the Framework in Place

Nesting Sprints in to Waterfall Frameworks - WHY

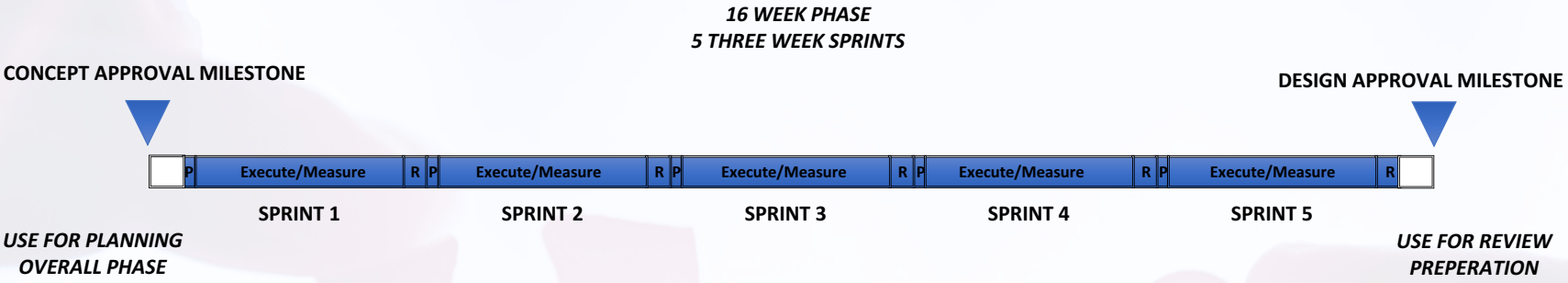


- Why Nest?
  - **For many tangible products, substantial qualification is required**
    - Hence the need for a testing phase
  - **For most, if not all tangible products, require substantial investment in tooling and inventory**
    - Hence the need for a milestone that ensures that we are prepared to do so
  - **For all tangible products, a comprehensive schedule is required to ensure that launch requirements are satisfied**
    - Hence the need for an end to end project plan
  - **Finally, for all tangible products, the notion of a MVP has limited applicability**
    - Hence the importance of a fixed and finite scope aligned with a timeline is required

Keep the Waterfall Framework in Place

Accelerates Agile Implementation – Needed for Testing & Financial Controls

Nesting Sprints in to Waterfall Frameworks - HOW



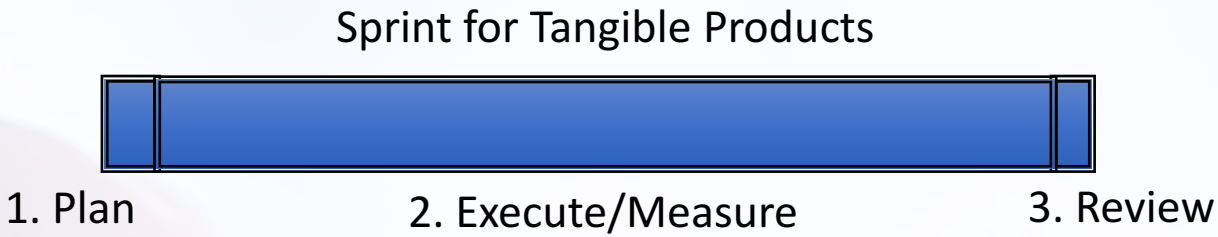
- How Nest?
  - **Most milestone processes have between 4 and 7 major milestones**
    - For the first and last phases, use 2 week sprints
    - For the middle phases, use 3 week sprints
  - **For all phases, keep the sprint length constant**
    - Why? Most importantly it keeps teams from extending (slipping sprints) and thus losing discipline

Sprints: Nuts and Bolts

- **Divide the planned phase period by 2 or 3**
- **Determine a whole number of sprints to be performed**
  - Start each sprint the first Monday after a milestone is passed
  - Use remainder to plan the next sprint, preparing for review, or to implement systematic improvements
- **Synchronize sprints with Waterfall Milestones**
  - By developing an overall project plan
- **How to we synchronize sprints with various Teams?**
  - By having the equivalent of a PMO (Project Management Office)
  - Can start without synchronized sprints (Crawl, walk, run)

16 WEEKS / 3 = 5.333 => 5 THREE WEEK SPRINTS EXTRA LEARNING AND PLANNING TIME (5 DAYS)

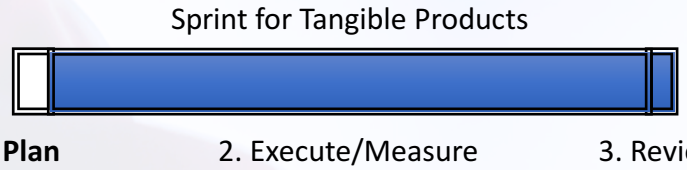
Three Components of Hardware Sprints



Three Components of a Hardware Sprint

1. **Planning (Typically a day)**
   - Goals are formulated
   - Estimated progress is described numerically
2. **Executing & Measuring**
   - Quantified progress is tracked visibly
   - Exceptions are managed
3. **Reviewing (Typically one half a day)**
   - Cross functional team reviews
   - Review of project & Review of process

Planning: Components of a Sprint



"CIRCLE DOT"

- **Define who does what**
  - Project Manager -> Scrum Master (Don't start with newbie)
  - Product Manager -> Product Owner (may need more staff)
- **Sprint Planning is different from Software**
  - Each Sprint is a bit unique – basic bones the same (Schedule, Deliverables, Metrics)
  - Details differ over arc of project – For example Sprints waiting for samples differ from in-house design Sprints
- **Create & Score Sprint Planning Template**
  - Helps clarify key components of Sprint – Purpose of Sprint 'n', the goal, and how you measure progress towards it

Sprint Planning Template

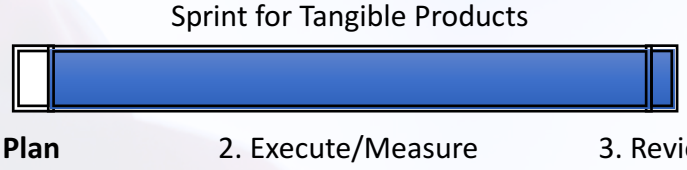
Sprint Planning Template – QUALITY SCORECARD (EXAMPLE)

5 = very clearly defined, 3 = average clarity, 1 = unclear

	Mean	Standard Deviation
Clarity of Design Intent	2.8	0.9
Risks/Key challenges Defined	3.7	0.9
Acceptance/Key deliverables Defined	2.9	1.5
Team/Responsibilities Defined	2.8	0.7
Suppliers/Partners Identified	4.3	0.5
Milestones Identified	3.9	0.6
Current Status	3.8	1.0
Specific Next Steps/Action Items	4.6	0.7
Expected Date of Delivery	4.2	0.7
	3.7	

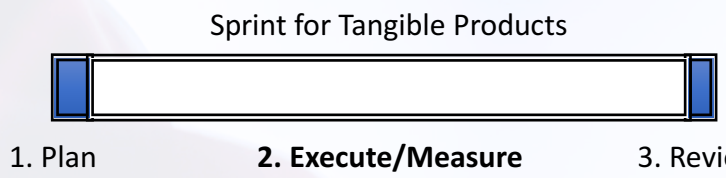
● *= Mean below 3, SD above 1: indicative of significant risk*

Planning: Measurement - Helpful rules of thumb Nuts and Bolts



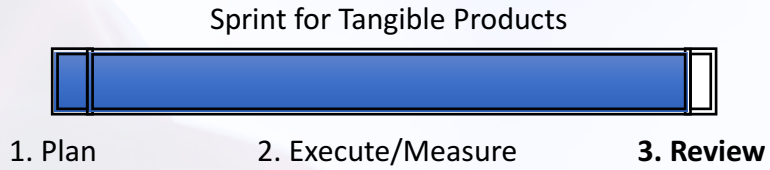
Metric Formulation Factors	Recommendations
• Keep the end in mind, what is the goal or Acceptance Criteria?	• Specify if a prototype, a stage of development, a financial review, a testing milestone…
• What are the key activities that lead to the goal?	• Prototype I, there could be many assembly/integration/test milestones; • Prototype II, based on planning poker, the team could estimate the sub-tax complexity • If a stage of development, there could be sum of PCB layout progress; % BOM specified. • If there is project tracking, you could measure Earned Value.
• Keep granularity small, are the metrics objective, easy to collect and display?	• Ideally you want metric systems where you would get at least one data point a day
• Define target curve – do you expect linear progress, or is progress uneven?	• If method for Prototype II is selected, with a sprint schedule, you could create a non-linear target curve • If method is tracking of many tasks and deliverable, a linear target can be employed

Executing/Measuring: Nuts and Bolts



- **"Daily" Standup Meetings**
  - Determined by team capability AND velocity of progress!
  - Can be helpful when there is an urgent problem OR when every day counts
  - Always BRIEF (LESS THAN 30 MIN)
- **Measure and post progress**
  - Plot progress over time
  - Agree on who collects and plots progress
  - Agree on what to do if progress is not sufficient
- **Latency issues are magnified with sprints**
  - One reason why Agile increases speed
- **Abandon the habit of serial prototype builds**
  - Prototype more frequently
  - Prototype in parallel (meaning start next build before prior build fully tested)
  - Anticipate and plan for a 50-100% higher prototype budget
  - Plan on doubling the number of prototypes

Review and Reflection: Components of a Sprint



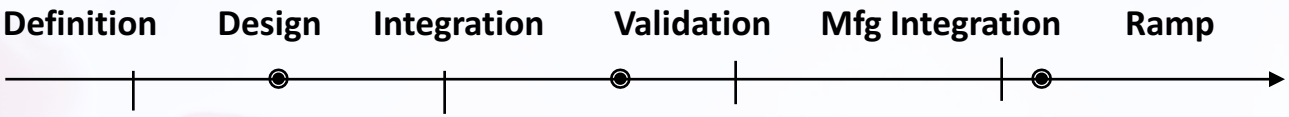
- **Review the execution of the PROJECT**
  - Take out the one pager and compare reality with the Sprint Plan
  - Did we accomplish our goal/meet our Acceptance Criteria?
  - What will we do differently for the next sprint?
  - Use the Project History technique (next page)
- **Review execution of the PROCESS**
  - How did our metrics work?
  - Were they easy to track? Meaningful? Correlate with true progress?
  - How did the team do? Do we have the right resources? Did we burn out the team?
  - What will we do differently for the next sprint?

Review and Reflection: Nuts and Bolts

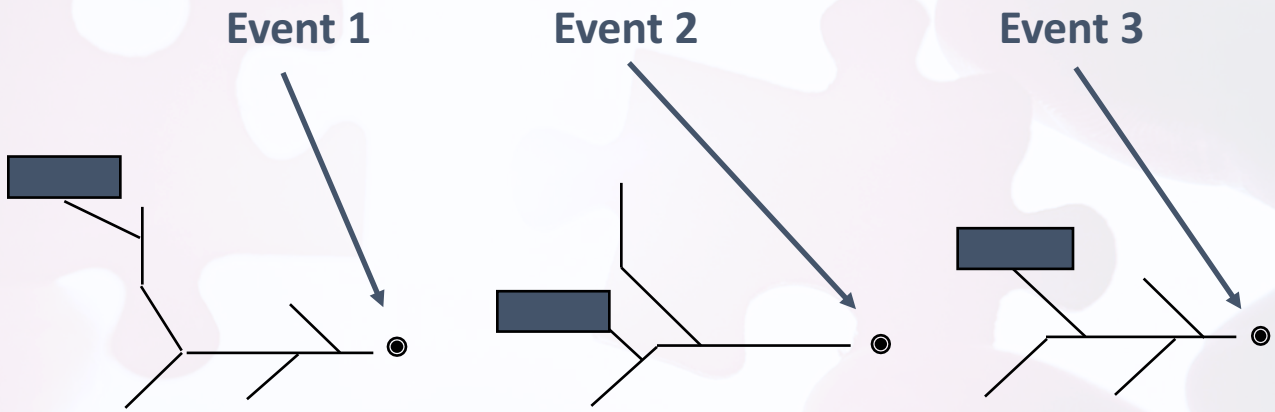
Sprint for Tangible Products

1. Plan 2. Execute/Measure **3. Review**

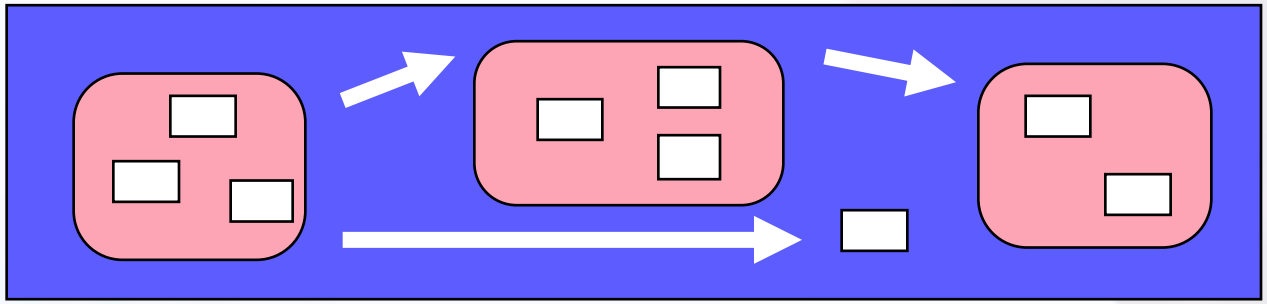
Event Analysis



Root Cause Analysis



Root Cause Synthesis



In Summary…

- **Highlights**
  - Better than either alone; AND not OR
  - Teams trump process
  - Agile can be done in five pieces
- **Questions?**
  - Waterfall AND/OR Agile?
  - Getting Started?
  - Organization?
  - Metrics?
  - Other?

Thank you!

Jeanne Bradford
jbradford@tcgen.com
408.828.5168

John Carter
jcarter@tcgen.com
650.283.9395

TCGen Inc.
Menlo Park, CA
www.tcgen.com