

Agile Project Management – Applying *Scrum* to Botball

David Mutchler

Professor of Computer Science and Software Engineering

Rose-Hulman Institute of Technology

5500 Wabash Ave.

Terre Haute, IN 47803

mutchler@rose-hulman.edu

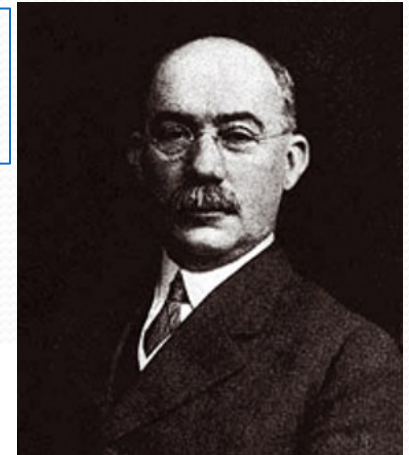
(812) 877-8426

Take-away (preview)

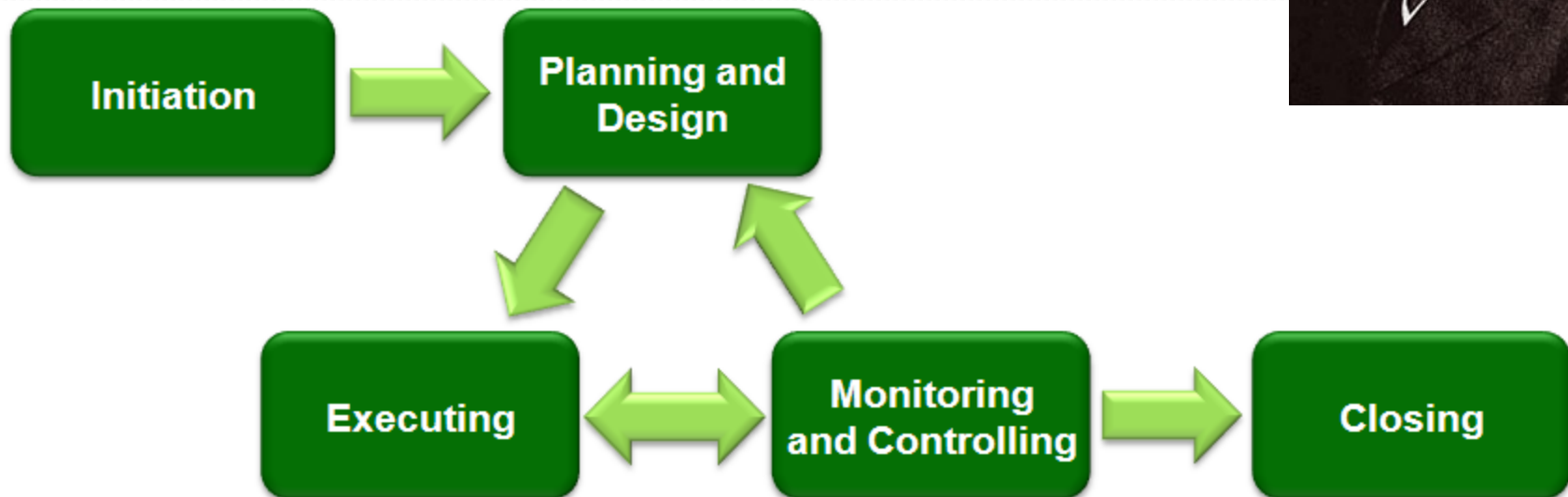
- Botball teams need Project Management
 - But Project Management is hard!
- Current focus:
 - Gantt Charts
 - Waterfall processes
- Another way:
 - *Agile* processes
 - **Simplified *Scrum* for Botball**

What is *Project Management*?

Henry Gantt (1861-1919),
the father of planning and
control techniques

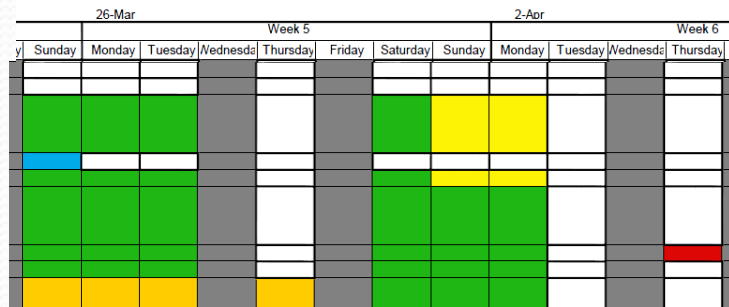
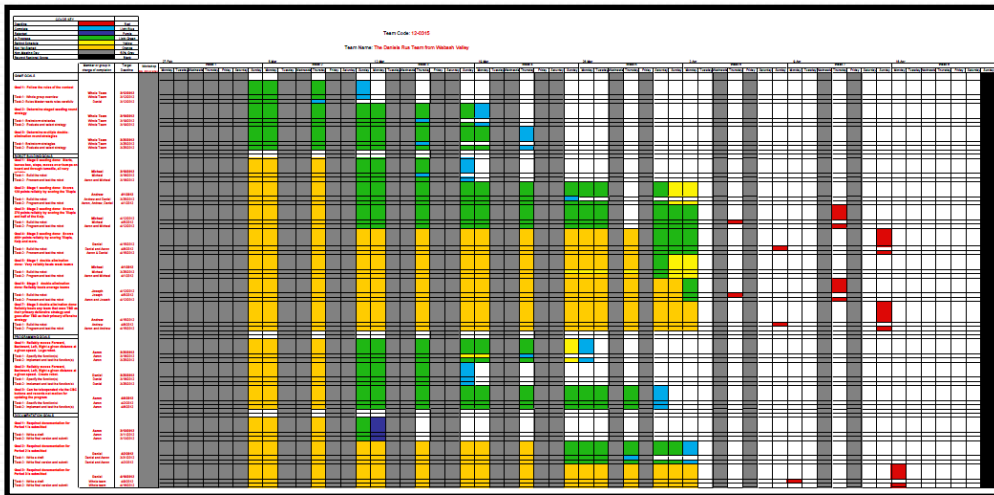


Typical phases of an
engineering project



Project Management in Botball –

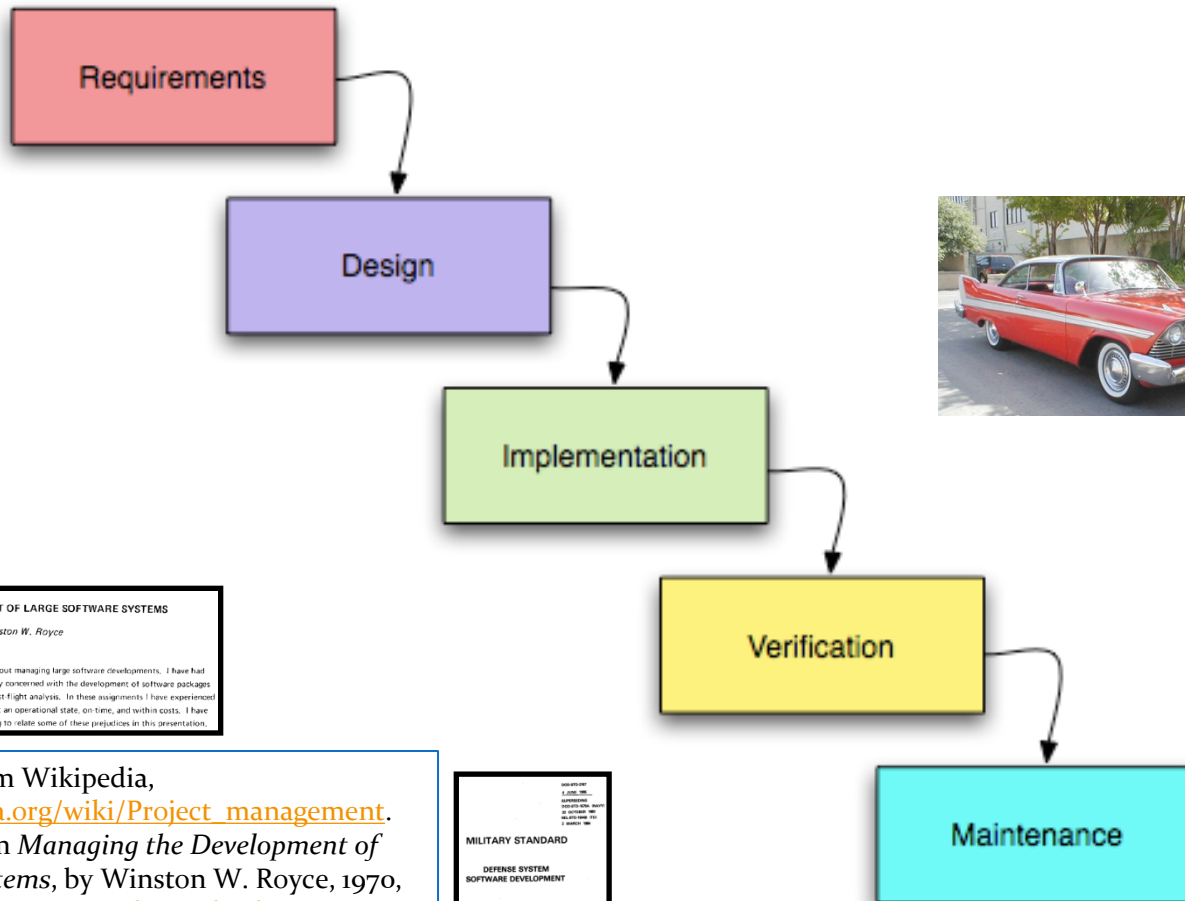
Gantt Chart



COLOR KEY		
Deadline		Red
Complete		Light Blue
Rejected		Purple
In Progress		Light Green
Behind Schedule		Yellow
Not Yet Started		Orange
Non-Meeting Day		50% Grey
Beyond Regional Scope		Black

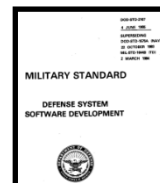
	Member or group in charge of completion	Target Deadline
GAME GOALS		
Goal 1: Follow the rules of the contest		
Task 1: Whole group overview	Whole Team	3/12/2012
Task 2: Rules Master reads rules carefully	Whole Team Daniel	3/12/2012
Goal 2: Determine staged seeding round strategy		
Task 1: Brainstorm strategies	Whole Team	3/19/2012
Task 2: Evaluate and select strategy	Whole Team	3/19/2012
Goal 3: Determine multiple double-elimination round strategies		
Task 1: Brainstorm strategies	Whole Team	3/25/2012
Task 2: Evaluate and select strategy	Whole Team	3/25/2012
ROBOT BUILDING GOALS		
Goal 1: Stage 0 seeding done: Starts, leaves box, stops, moves over bumps on board and through turnstile, all very reliable		
Task 1: Build the robot	Michael	3/18/2012
Task 2: Program and test the robot	Michael Aaron and Michael	3/18/2012
Goal 2: Stage 1 seeding done: Scores 120 points reliably by scoring the Tilapia		
Task 1: Build the robot	Andrew	4/1/2012
Task 2: Program and test the robot	Andrew and Daniel Aaron, Andrew, Daniel	3/25/2012 4/1/2012

Waterfall Project Management



MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS
Dr. Winston W. Royce

INTRODUCTION
I am going to describe my personal views about managing large software developments. I have had various assignments during the past thirty years, mostly concerned with the development of software packages for spacecraft mission planning, commanding and post flight analysis. In these assignments I have experienced different degrees of success with respect to arriving at an operational state, on time, and within costs. I have become prejudiced by my experiences and I am going to relate some of these prejudices in this presentation.



Building:

- Cars 😊
- Software ☹️

The diagram is from Wikipedia,
http://en.wikipedia.org/wiki/Project_management.
It was adapted from *Managing the Development of Large Software Systems*, by Winston W. Royce, 1970,
http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf.

Waterfall was incorporated in DOD standards like the one shown above,
<http://www.product-lifecycle-management.com/download/DOD-STD-2167A.pdf>

Agile Project Management

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

These sound like good fits for Botball!

We follow these principles:

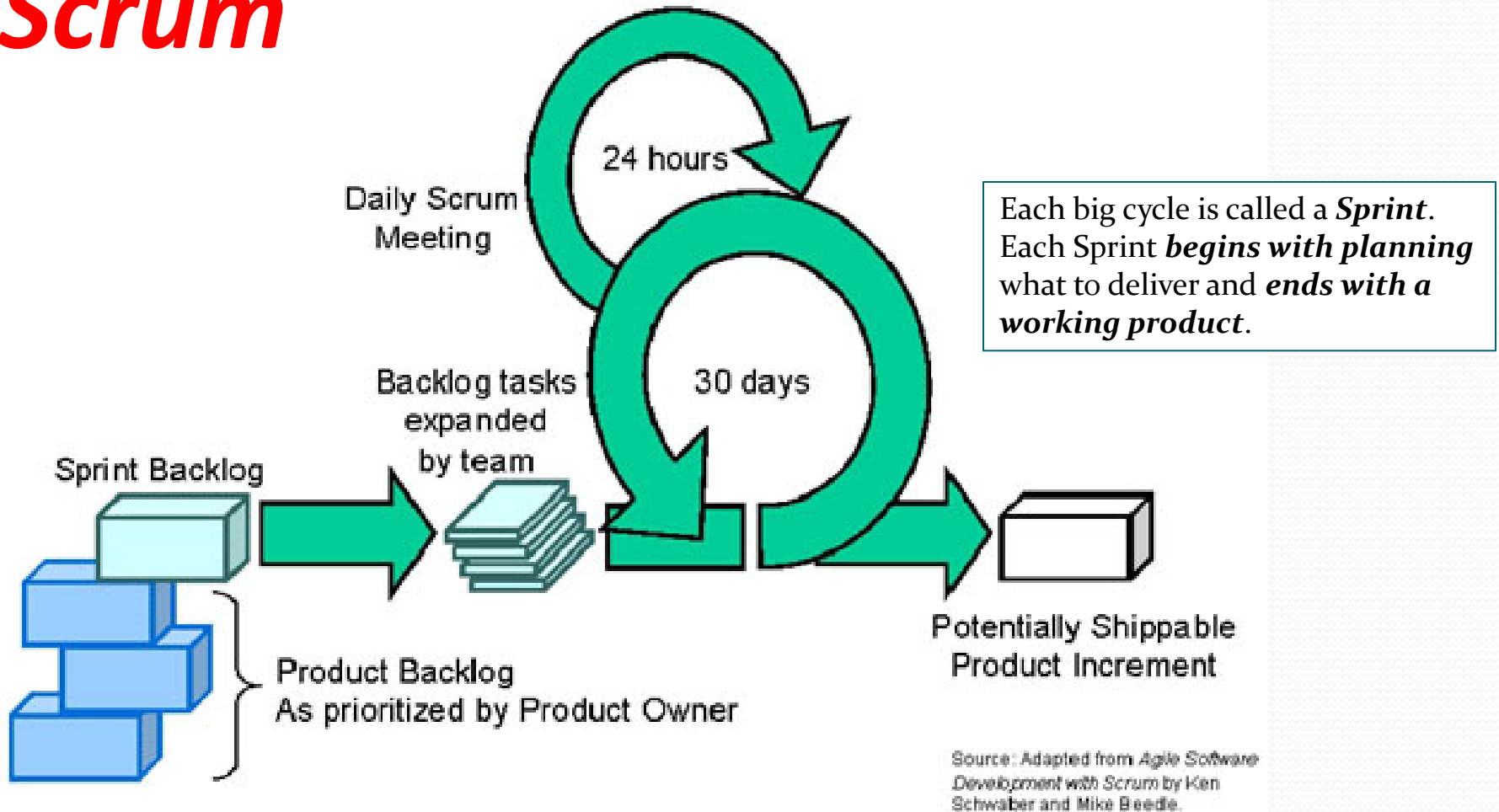
... **Deliver working software frequently** ...

... **Build projects around motivated individuals** ...

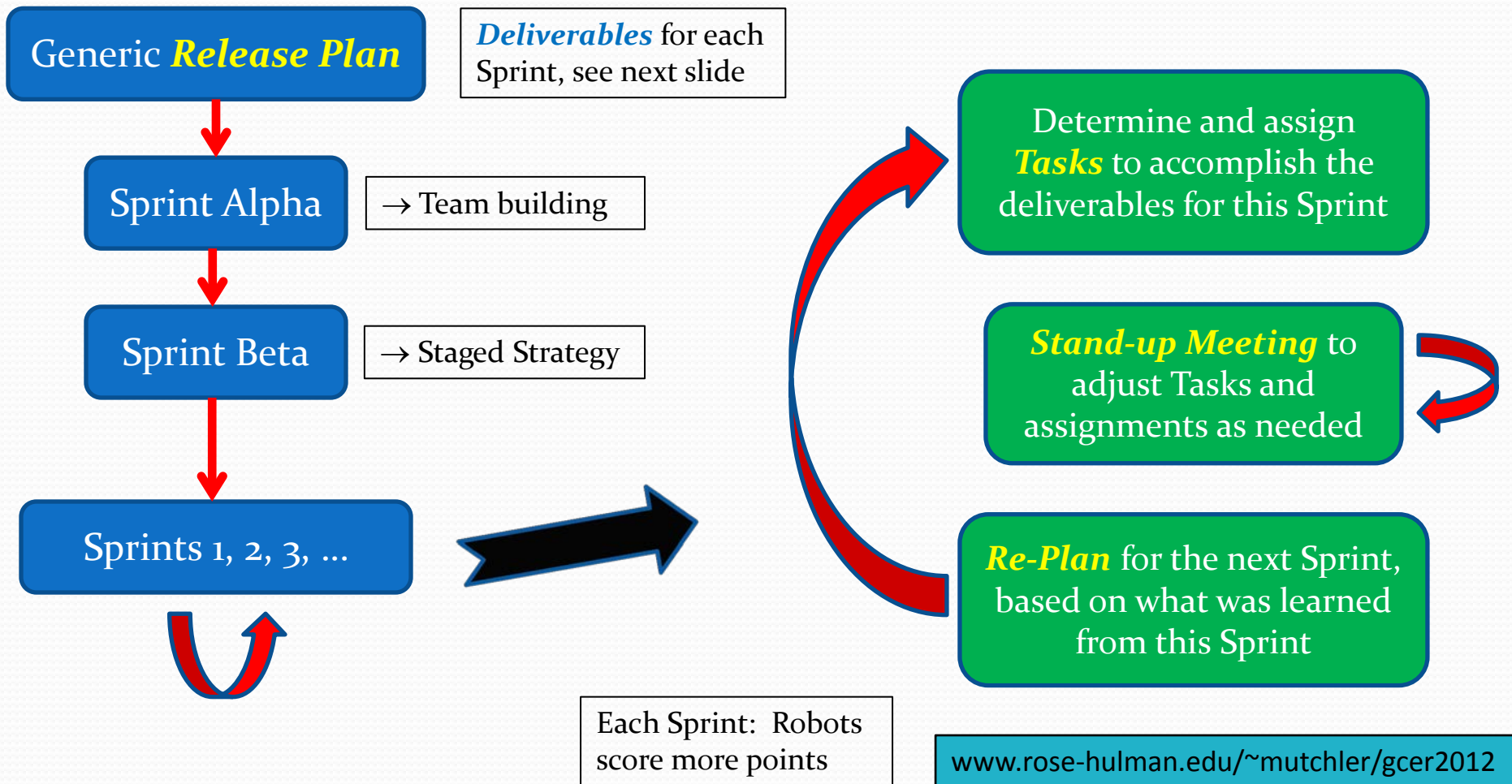
... **Working software is the primary measure of success** ...

... **Simplicity – the art of maximizing the amount of work not done – is essential** ...

Scrum



Simple Scrum in Botball



A generic *Release Plan* for Botball

- Sprint Alpha:
 - Plan for how to *build the team*
 - *Team rules*: participation, decision-making, conflict-handling, ...
- Sprint Beta:
 - *Staged Strategies* for Seeding and Double-Elimination
- Sprint 1:
 - Robots *accomplish Stage 1* of Seeding Strategy
 - Robots *accomplish Stage 1* of Double-Elimination Strategy
- Sprints 2, 3, 4...
 - Robots *accomplish Stage 2...* of Seeding Strategy
 - Robots *accomplish Stage 2...* of Double-Elimination Strategy

After Sprint Beta:

This and the next slides show an example of how the Release Plan might evolve. Only the Seeding Strategy is shown.

- Sprint 1:
 - Starts, leaves box, stops, moves over bumps on board and through turnstile, all very reliably.
- Sprint 2:
 - Scores $12 \times 10 = 120$ points reliably by scoring 12 of 12 Tilapia.
- Sprint 3:
 - Scores 180 additional points reliably by scoring 6 (or more) of 10 Kelp.
- Sprint 4:
 - Scores 410 additional points reliably by scoring 7 (or more) of 10 polyps from opponent's side and 3 (or more) from our side into reefs and into MPA.
- Sprint 5:
 - Scores TBD to TBD points reliably by scoring TBD.

After Sprint 1:

Sprint 1: Starts, leaves box, stops, moves over bumps on board and through turnstile, all very reliably.
Sprint 2: Scores $12 \times 10 = 120$ points reliably by scoring 12 of 12 Tilapia.
Sprint 3: Scores 180 additional points reliably by scoring 6 (or more) of 10 Kelp.
Sprint 4: Scores 410 additional points reliably by scoring 7 (or more) of 10 polyps from opponent's side and 3 (or more) from our side into reefs and into MPA.
Sprint 5: Scores TBD to TBD points reliably by scoring TBD.

- ***Accomplished:***
 - Starts, leaves box, stops, moves over bumps on board and through turnstile, all very reliably.
- Sprint 2:
 - Scores $12 \times 6 = 72$ points reliably by scoring ***at least 6 of 12*** Tilapia.
- Sprint 3:
 - Scores **60** additional points reliably by scoring **2** (or more) of 10 Kelp.
- Sprint 4:
 - ***Increases above scores for Tilapia and Kelp***
 - Scores 410 additional points reliably by scoring 7 (or more) of 10 polyps from opponent's side and 3 (or more) of 10 polyps from our side into reefs and into MPA.
- Sprint 5:
 - Scores TBD to TBD points reliably by scoring TBD.

After Sprint 2:

Accomplished: Starts, leaves box, stops, moves over bumps on board and through turnstile, all very reliably.

Sprint 2: Scores $12 \times 6 = 72$ points reliably by scoring **at least 6 of 12** Tilapia.

Sprint 3: Scores **60** additional points reliably by scoring **2** (or more) of 10 Kelp.

Sprint 4: **Increases above scores for Tilapia and Kelp.** Scores 410 additional points reliably by scoring 7 (or more) of 10 polyps from opponent's side and 3 (or more) of 10 polyps from our side into reefs and into MPA.

Sprint 5: Scores TBD to TBD points reliably by scoring TBD.

- **Accomplished:**

- Starts, leaves box, stops, moves over bumps on board and through turnstile, all very reliably.
- Scores $12 \times 3 = 36$ points reliably by scoring **at least 3 of 12** Tilapia.

- Sprint 3:

- Scores $12 \times 9 = 108$ points reliably by scoring **at least 9 of 12** Tilapia.
- Scores **60** additional points reliably by scoring **2** (or more) of 10 Kelp.

- Sprint 4:

- Scores **60** additional points reliably by scoring **4** (instead of just 2) or more of 10 Kelp.
- Scores **30** additional points reliably by scoring ~~7 (or more) of 10 polyps from opponent's side and~~ **3** (or more) of 10 polyps from our side ~~into reefs and~~ into MPA.

- Sprint 5:

- **Scores additional points from polyps.**
- Scores TBD to TBD points reliably by scoring TBD.

After Sprint 3:

Accomplished: Starts, leaves box, stops, moves over bumps on board and through turnstile, all very reliably. Scores $12 \times 3 = 36$ points reliably by scoring **at least 3 of 12** Tilapia.

Sprint 3: Scores $12 \times 9 = 108$ points reliably by scoring **at least 9 of 12** Tilapia.. Scores 60 additional points reliably by scoring 2 (or more) of 10 Kelp.

Sprint 4: Scores 60 additional points reliably by scoring 4 (instead of just 2) or more of 10 Kelp. Scores 30 additional points reliably by scoring ~~7 (or more) of 10 polyps from opponent's side and into reefs and~~ 3 (or more) of 10 polyps from our side into MPA.

Sprint 5: **Scores additional points from polyps.** Scores TBD to TBD points reliably by scoring TBD.

- **Accomplished:**
 - Starts, leaves box, stops, moves over bumps on board and through turnstile, all very reliably.
 - Scores $12 \times 11 = 132$ points reliably by scoring at least **11** of 12 Tilapia.
 - Scores **30** additional points reliably by scoring 3 (or more) of 10 polyps from our side into MPA.
- Sprint 4:
 - Scores **30** additional points reliably by scoring **1** (or more) of 10 Kelp.
 - Scores **50** additional points reliably by scoring **8 (instead of just 3)** of more of 10 polyps from our side into MPA.
- Sprint 5:
 - Scores **90** additional points reliably by scoring **4** (instead of just 1) or more of 10 Kelp.
 - Scores ...

Take-away (summary)

Consider *Simplified Scrum* (an *Agile* process)

- Make a Release Plan
 - KISS – don't overplan at this point
 - Items in the Release Plan are *Deliverables*, not Tasks. Most items should *score points*.
 - Repeatedly:
 - Do a Sprint
 - Do detailed planning (i.e., assign Tasks) *only* for the *current* Sprint
 - Evaluate
 - Re-Plan
 - Be agile! What the team has accomplished so far informs what the team can accomplish in the rest of the time!
- Always have working robots that score points

Acknowledgement: I am grateful to Sandy Mutchler for her help with the ideas of this project and this presentation of them.