Program Management

- Organizing and coordinating several projects’ results into one deliverable. That deliverable has the value to the organization
Program Management of Concurrent Projects

Program Management of Phased Projects
Issues in Program Management

- Especially in concurrent projects, how do you manage, what do you manage when you have 4, 7, 18, 25 teams of people all working on the same product?
- The issues are
  - Managing the risks, up, down, sideways
  - How to understand and explain status
  - How to manage the backlog
  - Architecture decisions

Activity

- We'll try a little project…
Agile Program Management

Agile Program Management Ain’t Easy

- Many people say, “Just do Scrum-of-Scrums”
  - How does that scale to more than 3 teams?
  - How does that scale to hardware?
  - How does that scale to phased releases?
  - Scrum-of-Sum has its place, and it’s not for everyone

Agile Provides Fast Feedback

- You know where you are every iteration
- Timeboxes provide urgency and focus
- It’s clear early if anyone is having trouble
- It’s clear early if the architecture is not going to work, assuming you use good tests

- Reduces overall project risk
Programs Are Riskier Than Projects

- You all know that projects don’t scale
- The larger and the longer the program, the more risky it is
- The more pieces the program has, the more risky it is
  - Hardware and software
  - Mechanical and hardware and software
  - Embedded and hardware and software
  - Regulated industries

Traditional Program Management

- Core team
  - Software program manager (person who program manages all the software teams)
  - Hardware program manager (person who program manages all the hardware teams)
  - Marketing program manager
  - Sales program manager
- Anyone else you need who can
  - Speak for that function
  - Commit people
  - Manage risks
  - Commit other resources
How Agile Changes Program Management

- Teams commit, not functional managers
- Product owners manage the what-to-build risk
- The teams manage
  - The how-do-we-build-our-features risk
  - Collecting and explaining their status
- Program management
  - Helps make the between-teams risks transparent
  - Collects and explains program status

Dirty Little Secret of Program Management

- If you’ve ever managed program successfully, you have used “agile” approaches:
  - Timeboxes to keep people focused
  - Implement by feature
  - Program-wide release criteria
  - Commit at the last responsible moment to high cost-of-change decisions
  - Prototype architecture
  - Interim milestones
One Approach to Programs I’ve Used for Decades (Not Agile)

Program Management, All Agile All the Time

- At the program level, need
  - Program manager
  - Overall architect
  - Overall product owner
  - Anyone else needed for the program (hardware, mechanical, finance, training, sales, support, …)

- For each team, need
  - Product owner, full time
  - Depending on how long the team has been agile, maybe a project manager/Scrum Master (< 2 years, yes!)
  - Architect, at least part time
All Agile Program’s Process

- Iteration 0
  - Develop the product backlog (Overall Product Owner with help from team product owners)
  - Charter the program (vision, product release criteria)
  - Any market-driven architecture decisions (Overall Architect with help from other architects, Sales, Support, Hardware, whomever)
    - Think Big Picture Up Front, not Big Design Up Front

For Each Iteration (Part 1)

- All start and end at the same time for each team (No staggered iterations)
  - Normal agile for the team
    - Each team selects its backlog items from the product backlog
    - Each team estimates and commits
    - Each team builds their iteration’s backlog, with the local architect as a regular member of the team
    - Daily standups
    - Each team elevates risks, issues through the PM/SM to the program team
    - Each team gets to release-able at the end of an iteration
      - If you have hardware, maybe demo-able
For Each Iteration (Part 2)

- Program team
  - Potential of a daily standup (think Scrum of Scrums)
  - Meets weekly to manage risks and figure out how to present status
    • Storyboards
    • Program velocity
- Architecture team
  - Uses what each team has learned to refine the architectural picture
  - May decide to hold architectural reviews periodically
    • To prevent technical debt, not to pre-define too far
  - Keep refining the Big Picture

Managing Status Up/Down

- This works for small programs, of up to, say, 7 teams. What about more teams?
  - There are too many people for a daily standup
  - Possibly too many risks for the program team to manage
- Need to collect software risks (or any other group of teams that’s relatively independent) as a delegate to the program team
Collecting Groups for Representation to the Program Team

- Several software reps to the program team
  - Gather some related teams and have them select one PM/SM to go to the program team
- One software rep to the program team
  - Organize the software like a program and have one overall software program manager
  - Each team sends a rep to the software program team and if too many, do some gathering
- Do some hierarchical gathering, based on the product

When You Have Hardware

- Nothing changes until you have pilot hardware
  - If the teams have only been getting to “demo-able” at the end of an iteration before pilot hardware, now they have to get to “releaseable”
  - The program will encounter problems
    - Be prepared to manage more risks
    - Be prepared to have sub-program meetings to manage the who’s going to do what in the hardware, firmware, software
Make the Program’s Progress Visible

- Agile makes the product visible as you proceed
  - You have to think about what you want for status for the program
  - Working product is the best
    - But you still need some data
    - Time burndowns are useless
    - Burnups might be helpful
    - Velocity might be ok, but velocity is personal for a team
      - You can’t add velocities together and have a program measure
  - Consider product backlog burndown

Progress Against the Product
Backlog (Burndown)
Program Management is Difficult

- Agile provides transparency
- Use a combination of techniques, and move to agile as you can

- No matter what lifecycle or combination of lifecycle you choose, make sure you:
  - Have interim milestones
  - Demo early and often
  - Raise risks and resolve them
  - Have fun!!
References and Reading

- Manage It! and Manage Your Project Portfolio have a number of how-to’s on programs
- I have a variety of articles on my website; from my email newsletter, the Pragmatic Manager; and in my Managing Product Development blog:
  - Pragmatic Manager email newsletter: www.jrothman.com/pragmaticmanager
  - If you’d like me to stay in touch with you, give me your business card or fill out a yellow sheet
  - Blog: www.jrothman.com/blog/mpd