



Splitting The Unsplittable

Techniques for breaking big work into
small increments for iterative
development

Michael James

Danube Technologies, Inc.

michael@danube.com



About Us

- Project success company
 - training on Scrum, Agile Engineering Practices
 - onsite consulting, mentoring, pair programming
 - free Agile PM tool: ScrumWorks
 - eat our own dog food
- Represented today: Kane Mar, Dan Rawsthorne, Victor Szalvay, Michael James
- Michael's background:
 - embedded systems for aircraft/spacecraft
 - J2EE/J2SE applications
 - “anti-architect”



Brief Presentation, Then Your Challenges

- Start thinking of your challenges now!



Benefits of Small Product Backlog Items

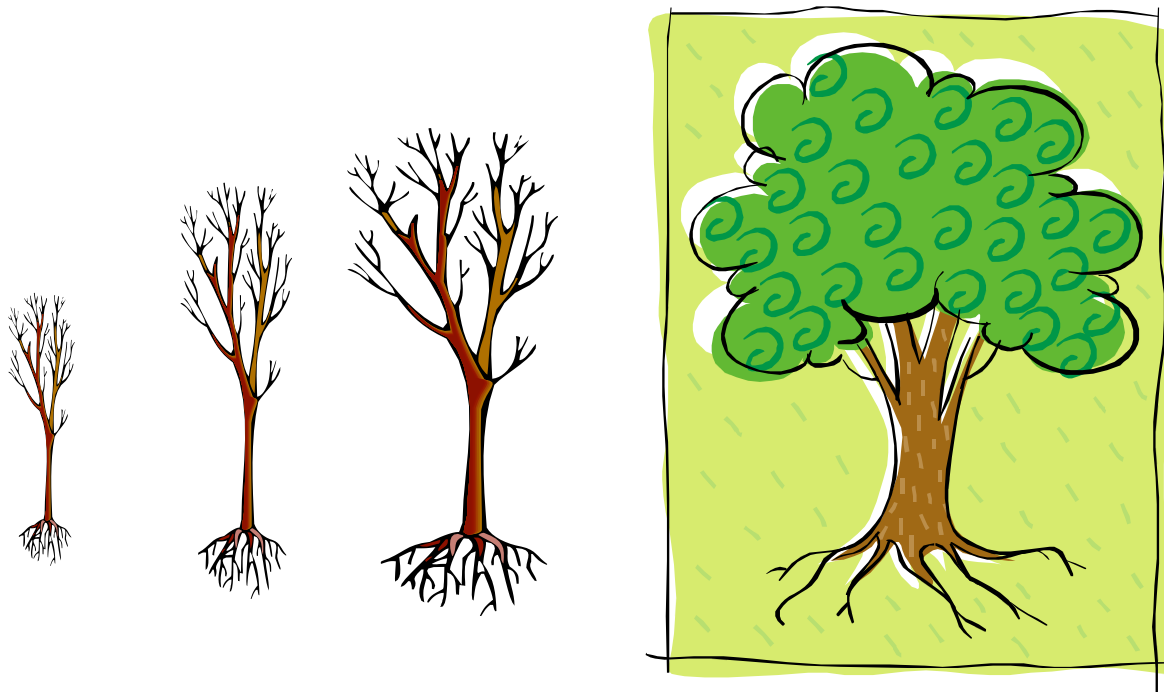
- Baby steps get you unstuck
- Easier to assess progress
- Tighter feedback loop with Product Owner/Stakeholders
- Finer-grained prioritization
- More fun to see working product sooner, even if initially “feature poor”



When to break them down?

- Just in time

Shoot For Thin Vertical Slices





Thin Vertical Slices

- Not Recommended: Online Store team backlog items “database logical model”, “database physical model.”
 - Challenge: Database schemas more expensive to change after release.
- Recommended: Small but visible functionality as underlying design emerges over time
 - Example: J2EE + Swing GUI client/server application with one dialog box.



Build Specific Solution before General Solution

- Not Recommended: Cell phone location-based services architecture that handles every kind of service, yet still must scale.
- Recommended: Build specific services actually needed by end user, then *refactor* to consolidate commonalities into emerging architecture.



Build Specific Solution before General Solution

- Not Recommended: Many sprints to build extensive clipboard infrastructure initially handling text, graphics.
- Recommended: Get user-visible cut/paste working for text, then for graphics, in different Sprints (iterations).



Build Specific Solution before General Solution

- Not Recommended: Premature internationalization.
 - *But consider whether Stakeholder acceptance criteria require internationalization/localization early.*



Build Static before Dynamic

- Not Recommended: School Information Management System with “everything” user configurable.
- Recommended: Start by hard coding, then build in configurability as actually needed.

Propagate Exceptions At First

- ❑ Not Recommended: Premature custom error handling in every “try/catch” block.
- ❑ Not Recommended: Swallowing exceptions (empty “catch” blocks).
- ❑ Recommended: Propagate Exceptions to outer layer (e.g. Front Controller). In future sprints, add custom handlers for exceptions that actually occur.



Make it Ugly At First

- “Make it work, make it fast, make it pretty”
– Dan Rawsthorne
- Example: SAT test preparation application initially works with ugly windows. Text replaced with graphics over time.



Make it Slow at First

- Not recommended: *premature optimization*
- Recommended: Build working application the simplest practical way, then use load testing and profiling tools to empirically identify actual bottlenecks.
 - Example: ScrumWorks database



Splitting *Tasks* Is Also Important

- Not recommended: 40-hour tasks
- Recommended: 1-day tasks
- Why?



Your Turn!

- What challenges are you facing with splitting stories or Product Backlog Items?

Other Resources

- <http://xp123.com/xplor/xp0512/index.shtml>
- http://danube.com/task_granularity
- http://danube.com/blog/kanemar/writing_user_stories.html