Parallel Agile Software Development

Barry Boehm, USC
GSAW STEM Session
February 27, 2019
Big projects don't add more sprints, they add more developers working in parallel.

Small, medium, (reasonably) large projects take the same amount of time (roughly 3 months) if enough developers are available

- No, we can’t do an entire crisis management system in 3 months

Merge and integrate at the end of each phase

Test team works concurrently with developers for each phase

Scale by adding developers, not stretching the calendar

Sequential Agile proceeds in a series of 2-week sprints. Bigger projects require more sprints and take longer to complete.

Parallel Agile proceeds in a series of three (roughly) month-long development efforts. Bigger projects require more developers but development time remains at approximately 3 months total.
Get to market faster without sacrificing quality

• 3 phases: Proof of concept, MVP, Initial Release
  – Each phase approximately a month long
  – Proof of concept uses storyboarding, prototyping to discover requirements, reduce risk
  – MVP uses UML modeling, details sunny/rainy day scenarios, reduce technical debt
  – Initial Release focuses on automated code generation, acceptance testing, performance tuning, optimization, reduce hotfixes
Database access code doesn’t get written manually

in round numbers this might be 20-40% of your code
Current status

- 2014-2015 Location Based Advertising (75 students)
  • Implemented commercially; discontinued due to low sales
- 2015 Picture Sharing (12 students)
  • Experiment comparison with Architected Agile project
  • PA project faster, less effort; comparable performance
- 2016-2018 CarmaCam (75 students)
  • In LA-Metro experimental use for bus-lane monitoring
  • Several additional organizations, applications interested
- 2017-2018 TikiMan Go Game project (25 students)
  • Being prepared for commercial application
Large Scale PA Critical Success Factors

- Three Team approach; similar to Bosch ART approach
  - Agile Rebaselining: Keeper Of The Project Vision/Architecture
    - USC: Rosenberg: Ensure MVC compliance, rainy-day use cases
    - TRW: Systems Engineering team; Handle all concurrency
  - Developers and Product Owners:
    - Rapid concurrent development
  - Independent Verification and Validation
    - Continuous across development
TRW Large-Scale PA Experience

- Walker Royce: 1-million SLOC Command-Control System
- Extensive early architecture and risk resolution; all concurrency done by 10 experts
  - 47 sequential-Ada programmers; Executing Arch. Skeleton
- Neil Siegel: several even-larger systems
  - Very high productivity; low error rate
  - Proof of value: worse productivity, error rate when new customer forced traditional approach; full productivity resumed when original approach resumed
• Masters students
  – Learn about new technologies
  – Learn how to apply them

• PhD students
  – Learn how to create new technology
  – Learn how to test hypotheses about its effects
STEM Aspects

• Computer Science
  – Model-View-Controller architecture

• Software Technology
  – Automatic code generation

• Software Engineering
  – Storyboarding, prototyping, 3-team approach

• Mathematics
  – Statistically-based cost estimation model