



A Quick Tour of Software Engineering

Gregg Vesonder
AT&T Labs - Research
vesonder@mac.com



Roadmap

- About me
- What is Software Engineering
- How does it differ from development/programming?
- A quick tour of software engineering
- Software Engineering and You
- Software Engineering Resources



Vesonder's Relevant Bio

- Software for 30+ years
- PhD in Cognitive Psychology - Computer modeling of learning and memory
- [Bell|AT&T] labs for 25+ years
- Dozens of projects AI, Data mining, Languages, Systems
- Reviewer of projects
- Stevens: CS540 (web version too), CS565
- University of Pennsylvania EMTM Program (Executive Master's in Technology Management)



Why Software Engineering?

http://www.standishgroup.com/sample_research/chaos_1994_1.php

- (in mid '90s) \$250B over 175,000 projects
 - Average cost of projects: large company \$2.3M, medium \$1.3M small \$.4M
- Almost a third will be cancelled before they are completed
- Over half will cost 189% of their current estimates
- Only 16.2% will be finished on time and on budget
- IBM FAA Air Traffic Control project -Mars probe ...
- From paint to building -- long life span
- the University of Oxford has this remark in their Engineering Science description, "The qualities of a good engineer include not only a high degree of technical competence but also imagination, strength of purpose, commonsense - and a social conscience."



Beginnings

- In late '60s the software crisis
- NATO conferences in 1968 and 1969 established the term software engineering
- See blog for url that provides the proceedings of the conference



Views of Software Engineering

- SEI:
 - Engineering is the systematic application of scientific knowledge in creating and building cost-effective solutions to practical problems in the service of mankind.
 - Software Engineering is that form of engineering that applies the principles of computer science and mathematics to achieving cost-effective solutions to software problems.



Quantitative Software Engineering

- “Quantitative Software Engineering is an analytical approach to producing reliable software products within budget and on time” – Stevens program
- Which matches the IEEE definition:
 1. The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is the application of engineering to software
 2. The study of approaches as in (1)



Software Engineering Knowledge

- SWEBOK, Software Engineering Body of Knowledge:
 - Software requirements analysis
 - Software design
 - Software construction
 - Software testing
 - Software maintenance
 - Software configuration management
 - Software quality analysis
 - Software engineering management
 - Software engineering infrastructure
 - Software engineering process



Software Process Models

- The cost of constructing most software occurs during development (broadly defined, development is not equivalent to coding!) and not during production
- Process is a series of predictable steps, a roadmap

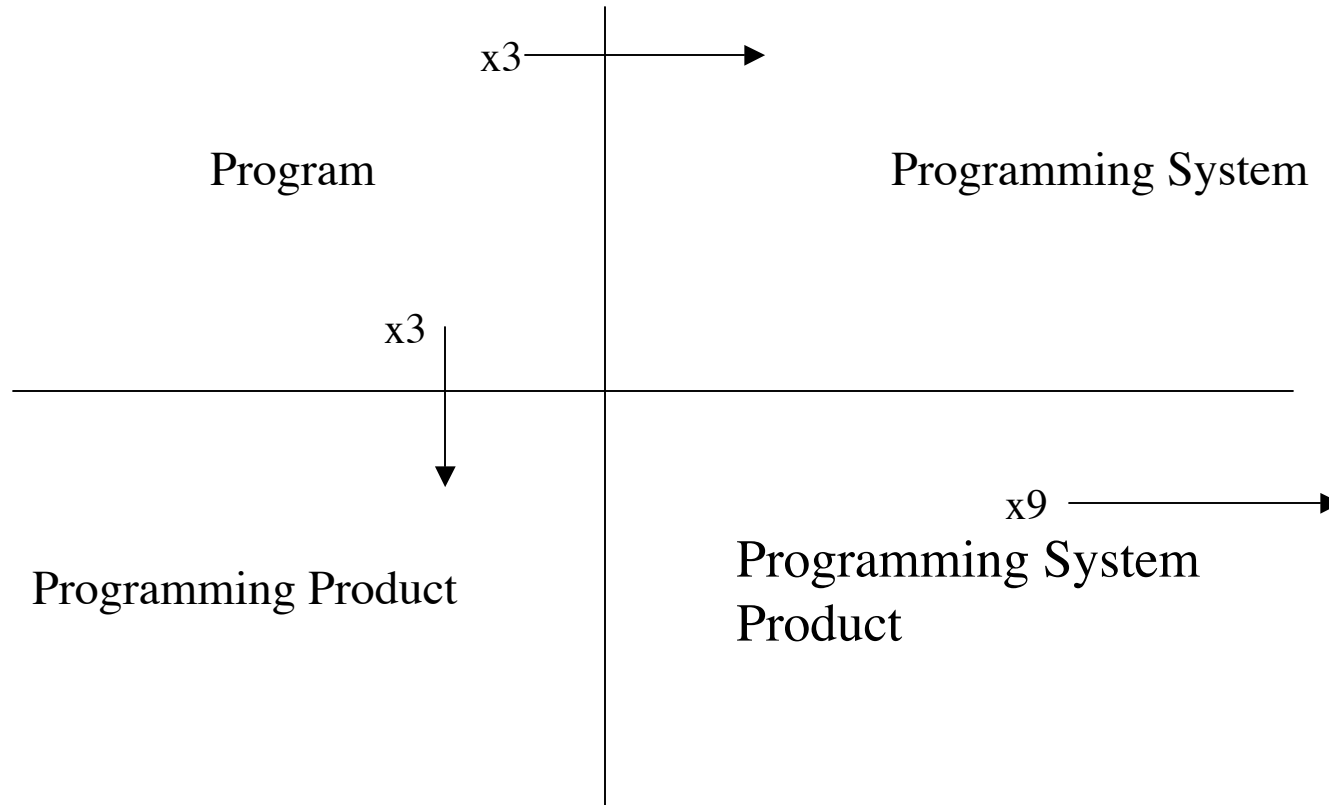


But First

- Code and Fix, Do Until Done Models
- No planning, general idea of product, informal “design” mostly through code
- Code, debug, test until ready for release
- No way to tell you are done or if requirements met

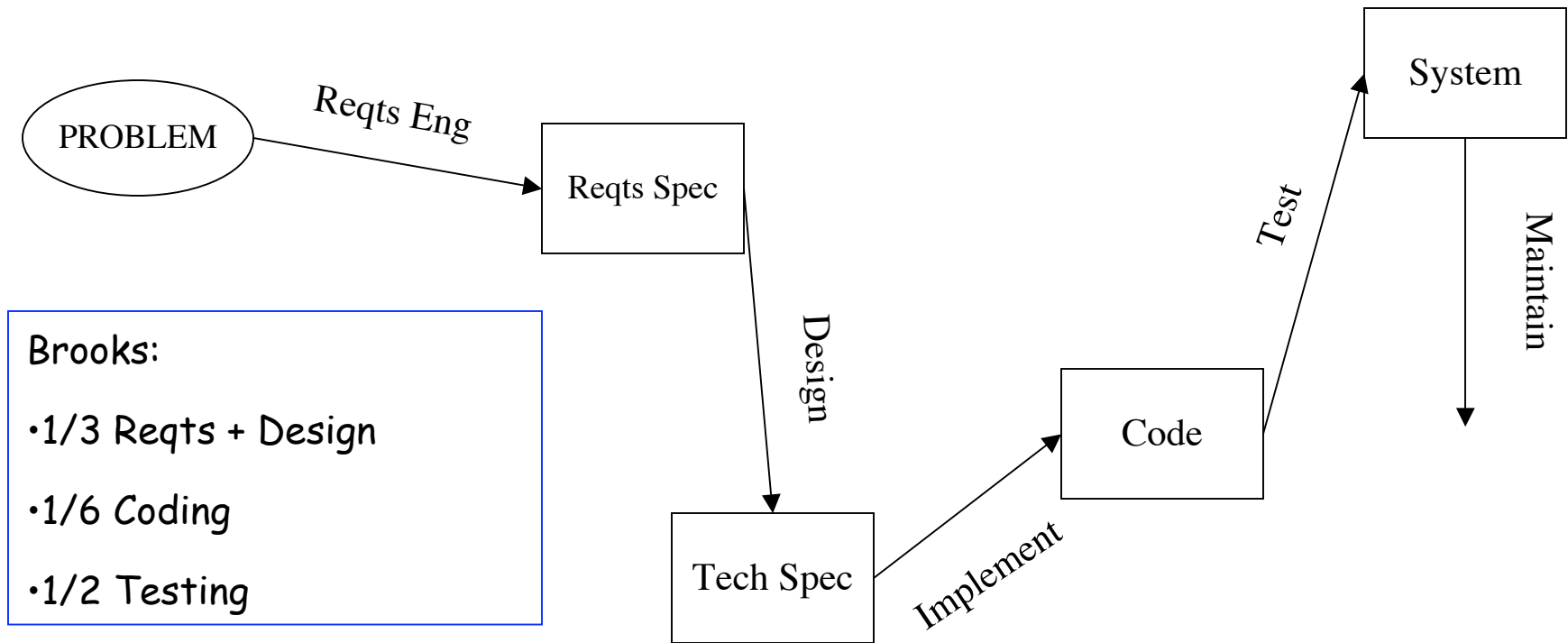


Brooks: System Production Differs from Coding





Simplified Model





Requirements

- Audience is “Stakeholders” and Designers – System Analysts
- What and Why not How – Domain Knowledge
- Functional and non-Functional (“ilities”)
- Requirements Elicitation
 - Interviews
 - Current System
 - Ethnography – side by sides
 - Prototyping – Danger
 - Story boarding, Documentation
- Rolling Stones – “You can’t always get what you want, but if you try sometimes well you might find you get what you need”
- Kobayashi Maru
- Consequences – errors introduced in Requirements are most expensive,



Design

- How
- Software architecture and software design
- Estimation - function points (Requirements stage too)
- OO methodology - CRC cards
- The UML (Unified Modeling Language)
- Cohesion and coupling



Code

- Usually the smallest percentage
- Mythical Man (Staff) Month - Brooks
 - Adding staff to a late software project makes it later
 - Termites not tornadoes - a project becomes late one day at a time.
- Unit Testing
- Documentation
- Code conventions: variable naming, spacing,
...



Test

- Quality
- Test Coverage Adequacy
- Mutation testing
- Test as much as you can afford
- Always squeezed
- Prevention - early involvement of testers



Types of Testing

- Unit testing - adjunct to coding, uses drivers and stubs, test cases source controlled
- Integration testing - test to uncover errors in interfacing
- Regression testing - subset of all tests to a given point to use when changes are made (part of build - smoke testing)
- Validation testing - succeeds when software functions in a manner that can be reasonably expected by the customer.. Alpha and beta testing are part of this
- System testing fully exercise the entire system:
 - Recovery testing - OA&M
 - Security testing
 - Stress testing
 - Performance testing
 - Reliability testing



Some Specialized Tests

- Testing GUIs
- Testing of Client server architectures
- Testing documentation and help facilities
- Testing real time systems
- Acceptance test
- Conformance test
- Your favorite here



Maintain

- Modification Requests (MRs) – bugs
 - Severity levels
 - Modification review board process
- Antiregressive maintenance
- Refactoring – OO based
- Forestalling Entropy
- 85% and up of original cost



And More

- Requirements, Architecture and Design Reviews
- Light Weight Process Models
 - eXtreme Programming, XP
- Computer Human Interaction (video - Knowledge Navigator)
 - <http://stevenf.com/mt/archives/000277.php>
- Open Systems - UNIX, PERL,...
- Outsourcing
- Project management
- Risk Management
- Formal Methods
- Game Development
- Reuse



Resources

- <http://homepage.mac.com/vesonder>
- <http://vesonder.typepad.com/universe/>
- <http://www.sei.cmu.edu/>
- My course and syllabus – follow along
- Books:
 - Mythical Man Month – Fred Brooks
 - Pragmatic Programmer – Andrew Hunt and Dave Thomas
 - Software Fundamentals – Hoffman and Weiss
 - Zen and the Art of Motorcycle Maintenance – Pirsig
 - Software Engineering – van Vliet



Software Engineering and You

- Team experience
- Development +
- Start now - discipline
 - Source code control
 - tests
- Internships
- Alpha hacker - join Open Source project