

18-642:

Key Development Metrics

10/16/2017



© 2017 Philip Koopman

“Without requirements and design,
programming is the art of adding
bugs to an empty text file.”

– Louis Srygley

Carnegie
Mellon
University

© 2017 Philip Koopman 1

Key Embedded Software Metrics

■ Anti-Patterns:

- **Development effort > validation effort**
- **Too many lines of code per hour**
- **Peer review finds <50% of all bugs**



■ Healthy project metrics:

- About 2 hours of validation effort per hour development
 - Tester:Developer head count ratio is about 1 to 1
- Productivity of 1-2 lines of code per hour for solid software
 - This includes entire process (requirements through acceptance test)
- Peer review should be finding >50% of all defects

Software = Design + Testing

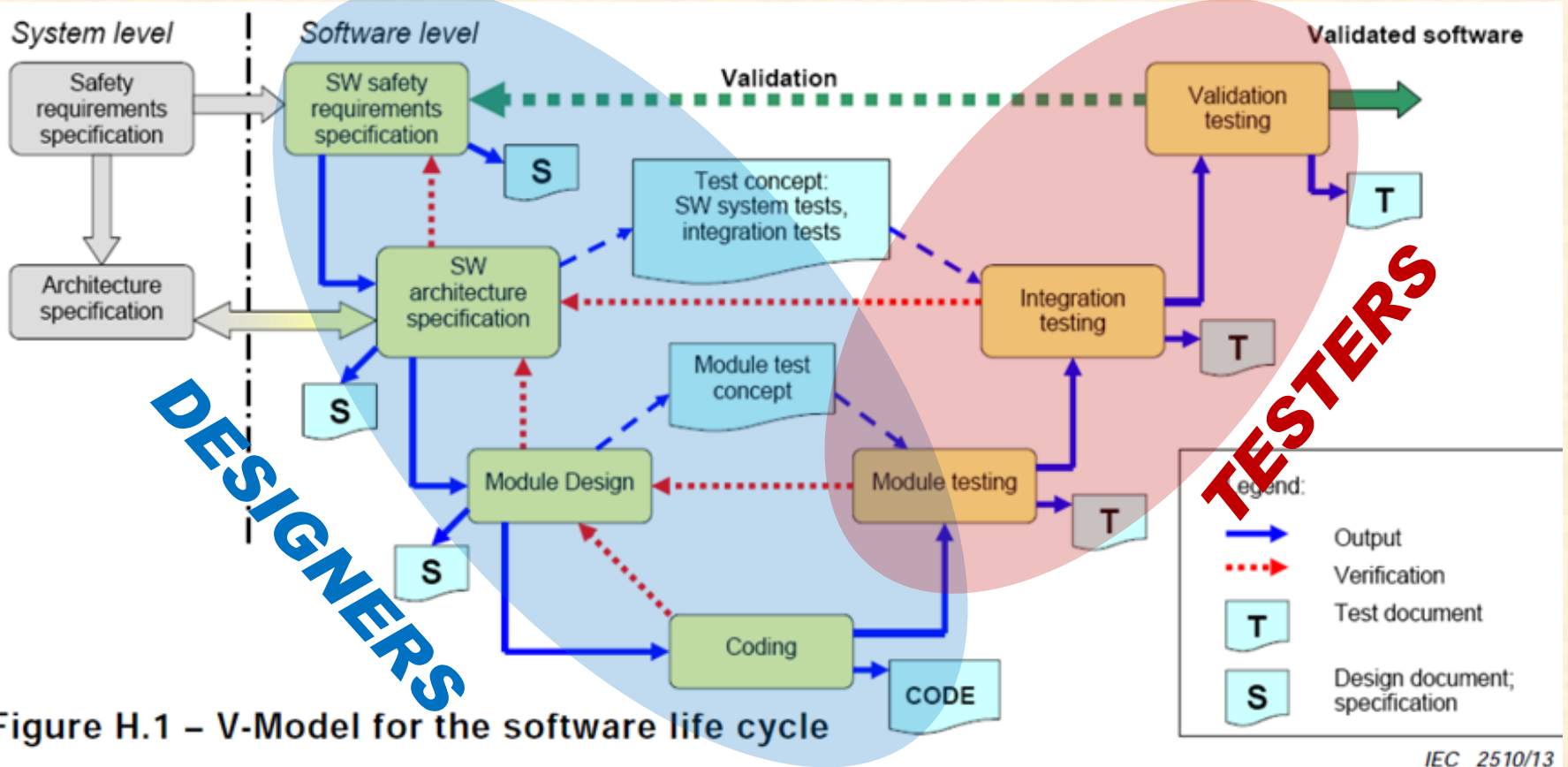


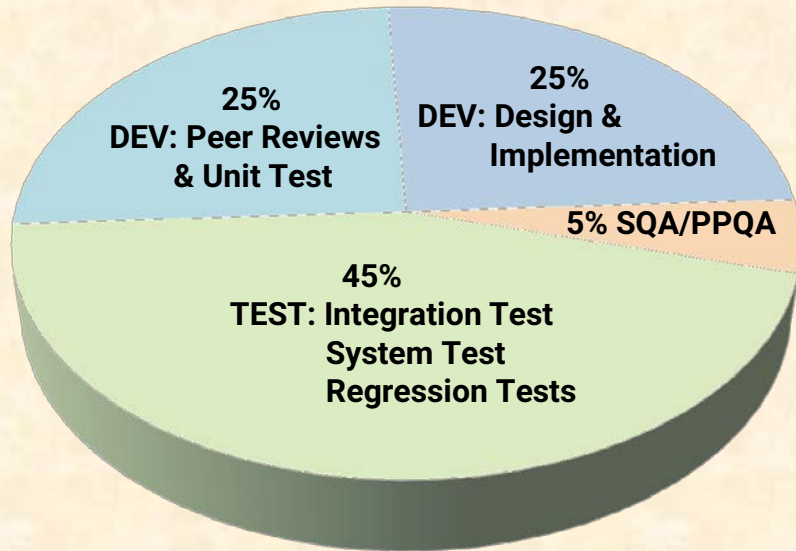
Figure H.1 – V-Model for the software life cycle

Typical Effort Distribution

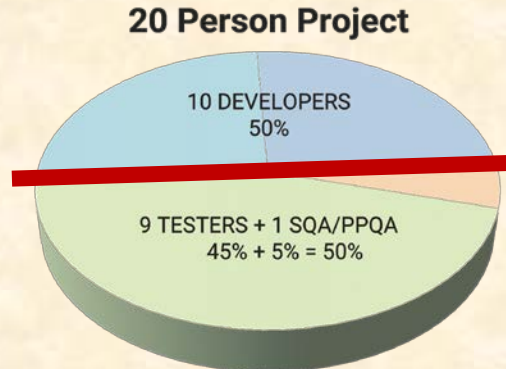
■ Tester to Developer ratio varies depending on situation

- Web development: 1 tester per 5-10 developers
- Microsoft: 1 tester per 1 developer
- Aircraft controls: ~5 testers per 1 developer

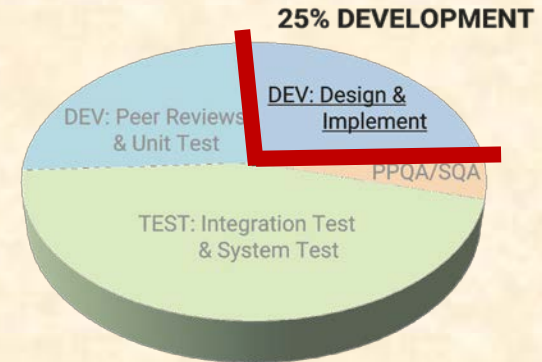
EMBEDDED SW PROJECT EFFORT



50%/50% Head Count



25%/75% Effort



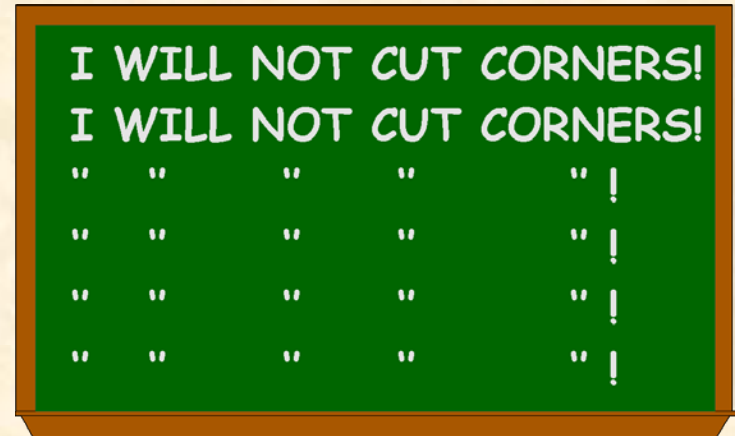
75% VALIDATION & QUALITY

Code Productivity

- **Productivity 1-2 lines of code/hr (including testers)**
 - Perhaps 3 lines/hr with Agile, but that speed increases quality risk

- **High lines of code/hr → cutting corners**

- Partial requirements, no design?
- No peer reviews?
- Only system level testing?



[Simpsons 7F11]

- **\$25-\$75 / line of source code**

- All-in cost, including entire V process, until field testing
- “Maintenance” can cost more, but might count as new project



Peer Review Effectiveness

- **Good peer reviews find 50%-70% of the defects**
 - Fewer than 40%-50% of defects found in peer reviews mean they are **BROKEN**
- **Peer Reviews cost perhaps 5%-10% of total project cost**
 - Let's do the math:
 - Peer reviews process about 100 lines of code per hour total
 - Three reviewers → 33 lines of code per person-hr
= 0.033 hours per line of code reviewed (2 minutes)
 - 0.033 hours review / .5 hours per LOC total = **6.7% for code review**
 - Plus review requirements & design ... but still a great ROI
- **Are peer reviews finding half your bugs?**
 - Are you spreading them out or bunching them together?
 - If they're not finding bugs, consider improving review culture



Best Practices For Key Software Metrics

- **2 hours of validation for each 1 hour of development**
 - Head count ratio generally 1 Tester to 1 Developer
 - About 5% of effort for SQA
- **Code productivity of about 1 to 3 lines per hour**
 - At or above 3 lines/hr, you probably are cutting corners
- **Peer reviews should find 50% (or more) of defects**
 - At about 5%-10% of total project cost
- **Metric Pitfalls**
 - Use only metrics that provide value – don't go crazy with metrics!
 - Gaming the metric doesn't improve software quality
 - Reward/punish based on metric values will render metric useless

