Access to the Experts

CUTTER CONSORTIUM

Investigating From Assessment to Reduction: Reining in Millions

John Heintz,

Owner of Gist Labs, Technical Consultant with Cutter Consortium

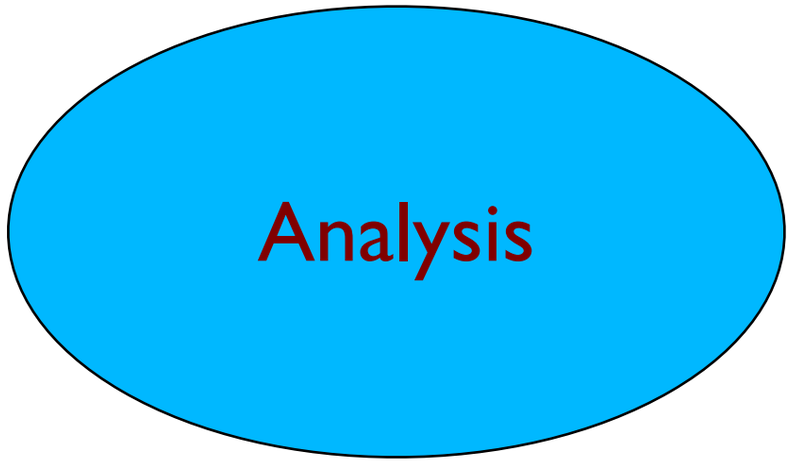
Israel Gat,

Director of Agile Product & Project Management Practice, Cutter Consortium

Second International Workshop on Managing Technical Debt
May 23rd, 2011

What Kind of Technical Debt?

- Automation Assisted Analysis
  - Complexity
  - Code Coverage
  - Rules, Violations, Lint
  - Copy/Paste
  - Public API Docs



- Can be automated into a Continuous Integration build
- Can be displayed in a Dashboard, and trended

The Commercial Context We Work Within

- A business **already struggling**
- Bugs, outages, failed releases
- Slow delivery of “anything”

http://www.flickr.com/photos/7682623@N02/4165393694/

Against the Grain

http://www.flickr.com/photos/7147684@N03/2154568117/

the developers are “stuck”

all new code “must” look like the old code

It took the technical debt SWAT team several months to get traction

reducing technical debt is against the grain within monolithic legacy code

In order to A, must do B, in order to B, must do C, ...

http://www.flickr.com/photos/austinevan/1225274637/

- Reducing tech debt **also means**
- training/mentoring
- unit testing
- refactoring
- design principles
- ...

- changing the working habits of many individuals

Example: Java Back-end Code

- Analysis showed $4.75/LoC
  - Added $2.4M from hand calculating lack of code coverage
  - Some code had been modularized, it had $3/LoC technical debt

Lines of code
1,206,590
1,648,693 lines
627,358 statements
7,414 files

Technical Debt
17.8%
$ 3,937,988
7,876 man days

Duplications
12.2%
200,783 lines
8,566 blocks
1,780 files

Complexity
4.8 / method
27.9 / class
33.0 / file
Total: 244,827

Violations
286,782

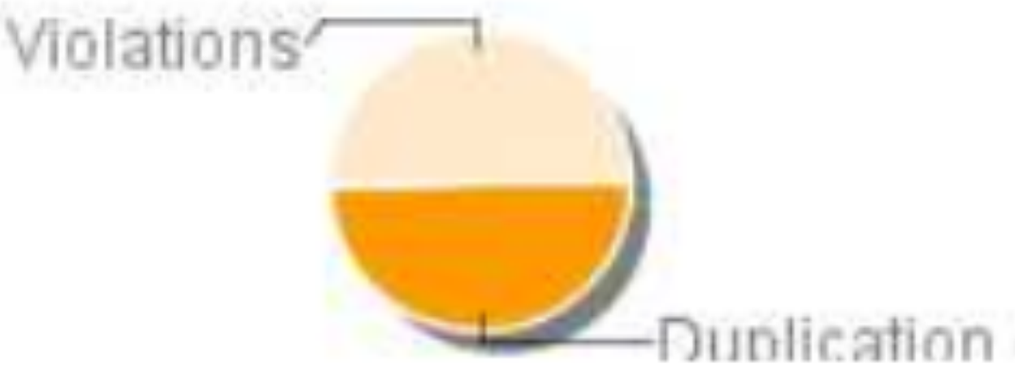
Blocker	0
Critical	1,621
Major	97,283
Minor	172,370
Info	15,508

Example: JSP Front-end Code

- Analysis showed $11.40/LoC technical debt
  - JSP Analysis currently only includes code violations and copy/paste

Lines of code
762,413
865,486 lines

Technical Debt
66.9%
$ 8,698,100
17,396 man days

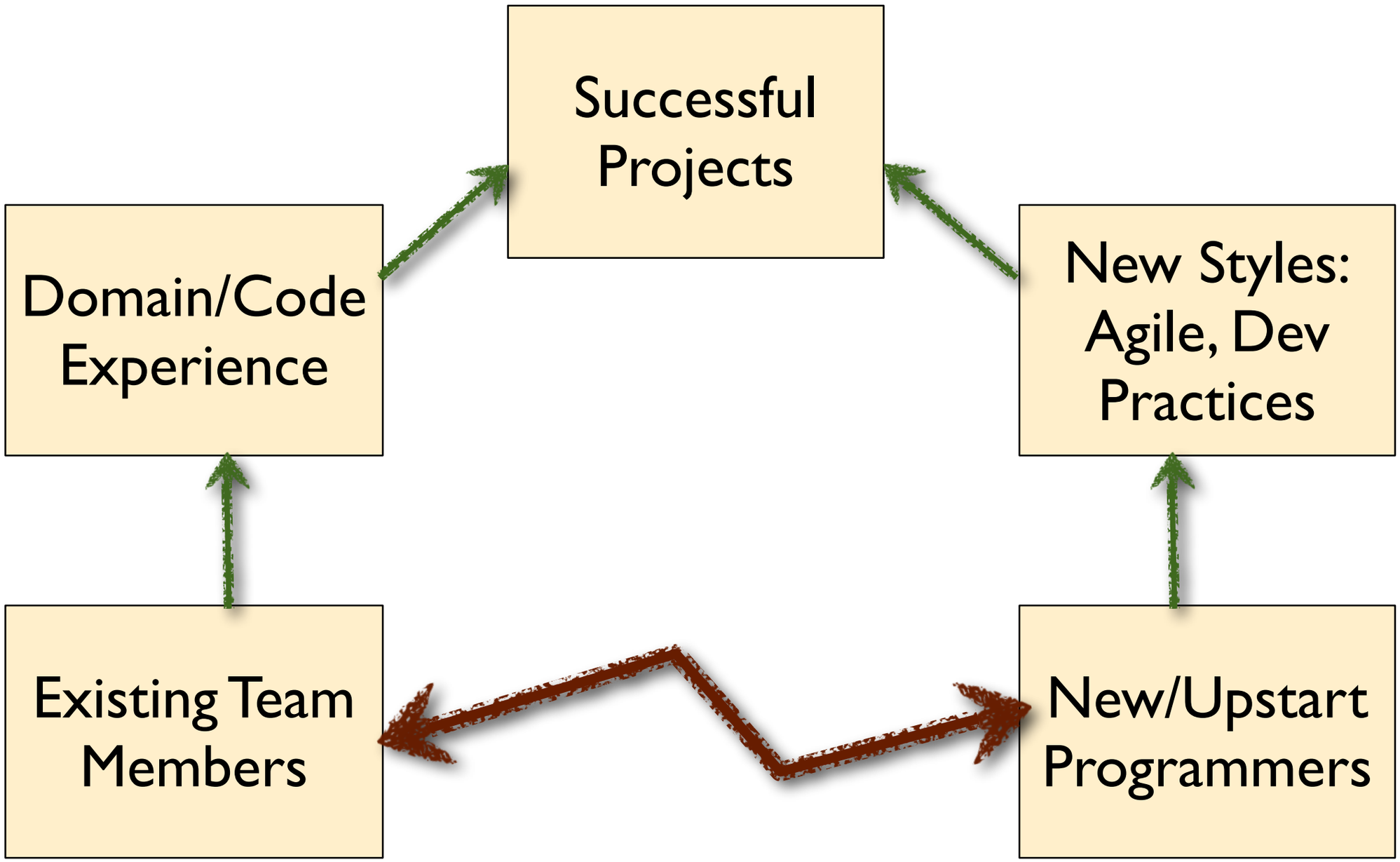


Duplications
40.3%
348,958 lines
34,500 blocks
2,679 files

Violations
701,696

Blocker	0
Critical	107,744
Major	177,962
Minor	415,990
Info	0

Conflict: Maintain what works, Change how we work



Progress So Far

- Analysis showed $4.3/LoC technical debt (down from $4.75/LoC)
  - Duplication and Complexity

Lines of code	Lines of code
1,206,590	**1,196,425** ▼
1,648,693 lines	1,613,686 lines ▼
627,358 statements	618,294 statements ▼
7,414 files	7,456 files ▼

Complexity	Complexity
4.8 / method	**4.8** /method
27.9 / class	**27.1** /class
33.0 / file	**32.4** /file
Total: 244,827	Total: 241,925 ▼

Duplications	Duplications
12.2%	**11.9%**
200,783 lines	192,062 lines ▼
8,566 blocks	8,984 blocks ▼
1,780 files	1,807 files ▼

Low Hanging Fruit, and Other Efforts

- Agile Rollout: To shorten product feature cycles
- The Technical Debt Agile SWAT team is focused on enabling Agile
- The measured reduction is from low hanging fruit:
  - Duplication and Complexity
- The unmeasured, but required, system changes include:
  - Build Script Fixes
  - Introducing Unit Testing infrastructure
  - Reworking the branch/merge version control policies
  - Further modularizing the system (partitioning the monolithic codebase)

Hard Work Yet To Come

- A team of “experts” have been working on the low hanging fruit
- More teams, with less training, are beginning to code on the “agile” branch
- more contention, more speed
- We anticipate progress to slow and the transition to increase stress (conflict) in parts of the organization
  - non-Agile teams
  - developers learning new rules

http://www.flickr.com/photos/arenamontanus/564129985/