

Software Engineering Introduction

CSCI 5828: Foundations of Software Engineering
Lecture 05

1

What is Software Engineering?

Solving problems with software-based systems

Design and development of these systems require -

- **Analysis**
 - decomposing large problems into smaller, understandable pieces
 - abstraction is the key
- **Synthesis**
 - building large software systems from smaller building blocks
 - composition is challenging

2

What is Software Engineering?

To aid us in solving problems, we apply techniques and tools

Techniques - a formal "recipe" for accomplishing a goal that is typically independent of the tools used

- automated builds, configuration management, software testing, etc.

Tools - an instrument or automated system for accomplishing something in a better way, where "better" can mean more efficient, more accurate, faster, etc.

- gradle, git, jenkins, etc.

3

What is Software Engineering?

To aid us in solving problems, we apply

Procedures - a combination of tools and techniques that, in concert, produce a particular product

Paradigms - a particular philosophy or approach for building a product

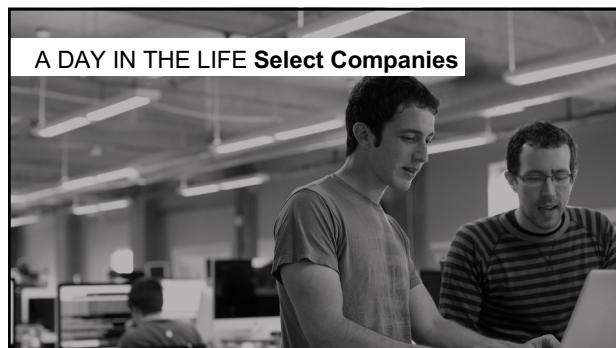
- Think: "cooking style": may share procedures, tools, and techniques with other styles but apply them in different ways

Both approaches use similar things -

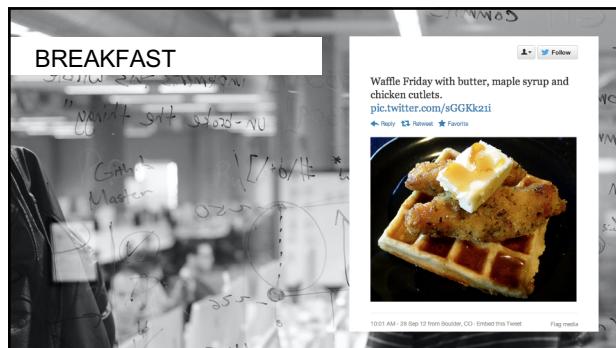
reqs., design, code, editors, compilers, etc.

But think about the problem in fundamentally different ways

4

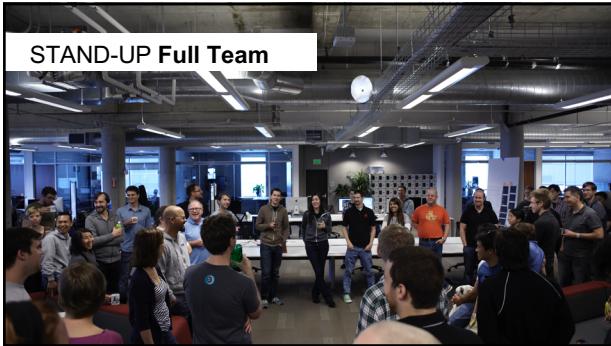


5



6

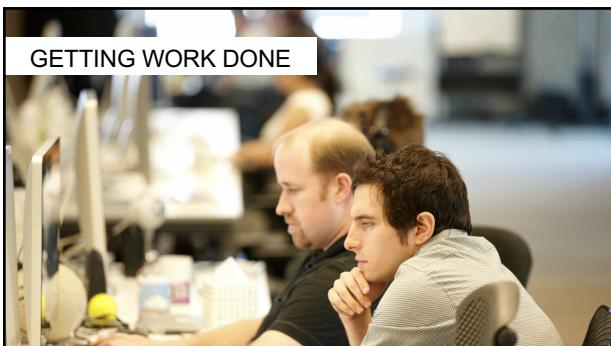
2



7



8



9



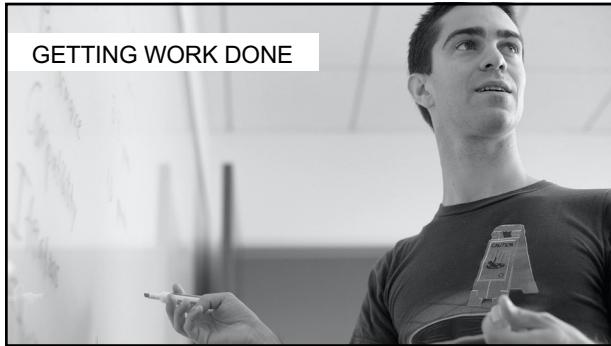
10



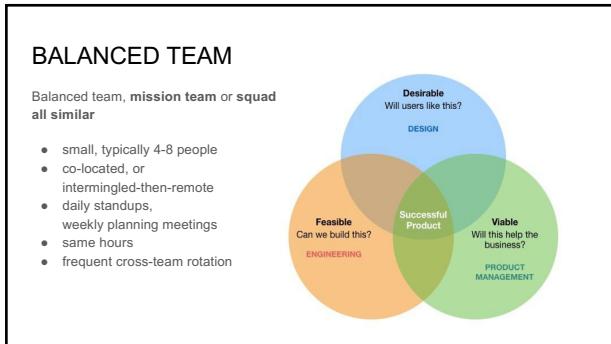
11



12



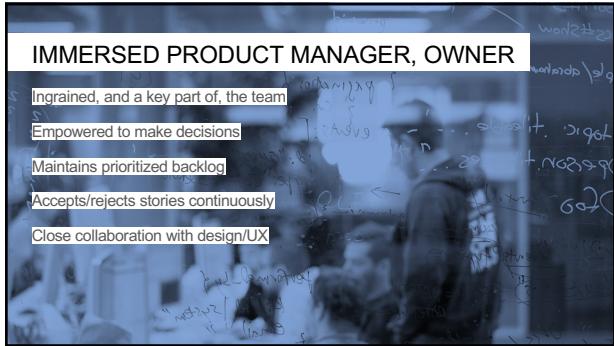
13



14

BUSINESS DRIVEN

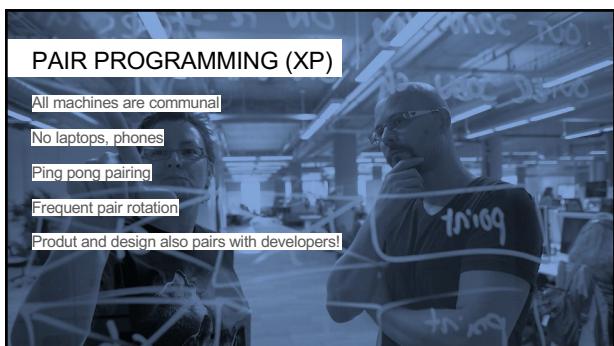
15



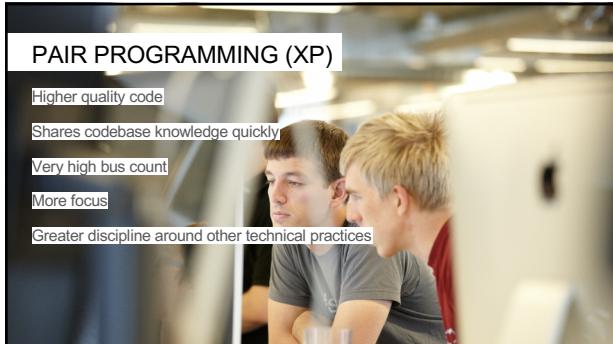
16



17



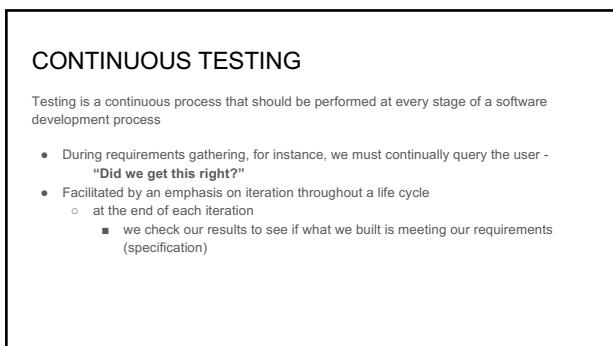
18



19



20



21

TESTING THE SYSTEM

Unit Tests

- Tests that cover low-level aspects of a system
 - For each module, does each operation perform as expected
 - For method foo(), we'd like to see another method testFoo()

Integration Tests

- Tests that check that modules work together in combination
- Most projects on schedule until they hit this point (MMM, Brooks)
 - All sorts of hidden assumptions are surfaced when code written by different developers are used in tandem
- Lack of integration testing has led to spectacular failures (Mars Polar Lander)

22

TESTING THE SYSTEM

Acceptance Tests

- Tests performed by the user (product owner) to check that the delivered system meets their needs
- Developers will be working directly with product owners and then respond to problems as they arise

23

TESTING THE SYSTEM

Once we have code, we can perform three types of tests

White Box Testing

- Since, we have access to most of the code, let's make sure we are covering all aspects of the code: statements, branches, ...

Grey Box Testing

- Having a bit of insight into the architecture of the system, does it behave as predicted by its specification

Black Box Testing

- Does the system behave as predicted by its specification

24

CODE COVERAGE

A criteria for knowing white box testing is "complete"

Statement coverage

- write tests until all statements have been executed

Branch coverage (aka edge coverage)

- write tests until each edge in a program's control flow graph has been executed at least once (covers true/false conditions)

Condition coverage

- like branch coverage but with more attention paid to the conditionals (if compound conditional, ensure that all combinations have been covered)

25

CODE COVERAGE

A criteria for knowing white box testing is "complete"

Path coverage

- write tests until all paths in a program's control flow graph have been executed multiple times as dictated by heuristics, e.g.,

for each loop, write a test case that executes the loop

- zero times (skips the loop)
- exactly one time
- more than once (exact number depends on context)

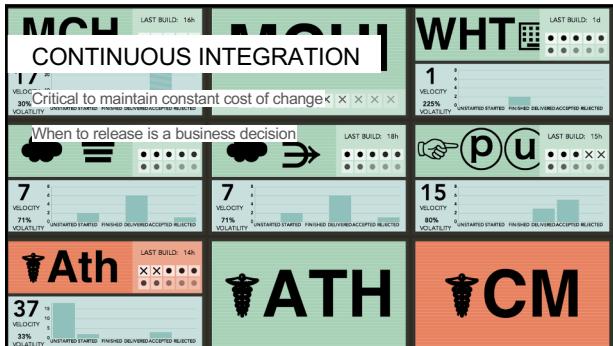
26

REFACTORING

Code evolution
Incremental design towards patterns
It takes discipline, but pairing helps



27



28

Pipelines CLUSTERS LOAD BALANCERS SECURITY GROUPS TASKS CONFIG

+ - Group by Pipeline Show 2 per group

BUILD #2 TRIGGERED BUILD Hello Jenkins! 2019-11-24 18:05:10 EST Status: RUNNING Duration: 00:12 Trigger Enabled Configure Start Manual Execution

Jenkins Bake Deploy Hide Details

JENKINS DETAILS Step Started Duration Status Jenkins Jenkins Config Task Status Jenkins Stage Configuration Jenkins Jenkins Job: Hello Build Build

CONTINUOUS DELIVERY

29



30

Questions?
