Continuous Delivery

software development & process & prototyping

[recall] spike driving machine & building a railroad

- what's available?
- what's the problem?
- Then, how?

[recall] incremental / evolutionary prototyping & process

- refine the product over time
- successive prototypes are constructed,
- released to clients for feedback.

problems

- what's the notion of a release?
- additional manual work for integration (integrate the code into a runnable prototype..)
- by the dev side: a prototype that functions well internally
  » it works for the client?
  - fully tested? e.g., for a certain device
  - any details overlooked?

to avoid these oversights?

automate the build and integration aspects of a project?

continuous delivery

solution: continuous delivery

deliver a product continuously, as it's being developed.

commiting a code change

- built
- tested
- integrated
- released

- problems can be noticed right away
- time between making a change and having it released can be very short

in continuous delivery

product can be released at any point [if you feel it's ready]

prototypes in channels

continuous delivery & prototypes in channels/streams

- place prototype releases in specific **channels**, or **streams** intended **for different audiences**
- » to make sure continuous releases are tested properly before being released to the **public**.
- Developers can gain insight into their product by receiving feedback from each channel.

different prototype channels

- a developer channel
  » for day to day builds developers generate (which is not ready for widespread use)
- a test channel
  » for prototypes created for a group internal to a company
- a stable channel
  » for releases targeted at the core users

continuous delivery

automated tools & continuous delivery

automated tools for supporting continuous delivery
» to build and integrate the code, run tests, and package the product into a releasable form..

test-driven development: best practice of automatic testing

- write tests before writing the code
- ensures that you're actually solving the right problem and making the functionality you want

continuous delivery

direct benefit

ensures that the process is happening **all the time**

- » if nothing breaks during the process, there should be a prototype ready for distribution at any time.
- » gathering feedback

continuous delivery & releasable prototype

aims to have a releasable prototype/product, at the end of every iteration.

- abandon your project?
- release the product without completing all the plan features if resources ran out?

with continuous delivery, product quality would actually improve

- in each iteration, make sure that everything works properly in small doses
- fix the problems occured when developpers try to build, test, integrate, and release one big product all at once..

Quiz

context

Howie and his development team are constructing the support infrastructure to enable the continuous delivery of the product prototypes put together by other developers. He has automated tools in place to

- build and integrate code,
- package the product,
- and install the product

in a test environment.

In this infrastructure, he also needs automated tools to do what?

- A. make prototypes.
- B. do detailed design.
- C. write the code.
- D. run tests.

Quiz

In this infrastructure, he also needs automated tools to do what?

- A. make prototypes.
- B. do detailed design.
- C. write the code.

✓ D. run tests.

a real example of continuous delivery 1/2

Microsoft Daily Build

- ensure that programmers are in sync at the beginning of each build activity
- following a process that makes developers integrate their code into the larger system
- uses a system of Continuous Integration to make sure that nobody wanders too far off the path

automatic testing to ensure that it will work with the project as a whole

- how their work fits into the project as a whole
- how their work affects other members of the team

a real example of continuous delivery 2/2

the effects

- keeps your developers' morale up
- increases the quality of the product
  by giving developers the ability to catch errors before they become a real problem.
  - a piece of code into the system, and it fails the tests
  - something integrated into the previous build cause the product does not work

Quiz

context

Thomas is a developer working for Microsoft. He has just spent his whole day writing the code, which will become part of the next version of the Windows operating system. At the end of the day, he uploads his changes onto a server.

What can cause his changes not to be tested?

- A. his code does not work,
- B. his code does not build,
- C. other code in the product does not work,
- D. other code in the product does not build.

Quiz

What can cause his changes not to be tested?

- A. his code does not work,
- ✓ B. his code does not build,
- C. other code in the product does not work,
- ✓ D. other code in the product does not build.

- build matters
- code that builds, but does not work, could still be tested

summary

continuous delivery

- used to release incremental or evolutionary prototypes
- can be incorporated into an iterative process, e.g., the unified
- example: Microsoft Daily Build

spike driving machine

- Unified process & prototyping + continuous delivery (require infrastructure) » for large, long-term projects, in which the product's quality could be severely affected by faulty changes made by the development team
- more robust than the Waterfall (),
  but sledgehammers are needed for small projects

On adapting processes

be open-minded, do/create what works best for you

- iterative or parallel software process models are the good/best and only tools ?! (for software product management)
- There are applications for each tool,
- -> it's up to you as a software product manager to apply them in the right situations.
- none of the processes mentioned throughout this module are necessarily entirely independent from one another.
- You can reasonably integrate aspects from each process to create your own.