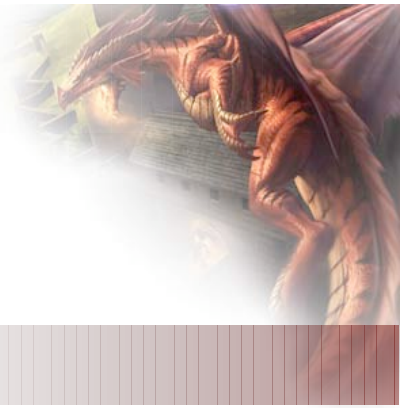
Agile Process

CMPUT 299

H. James Hoover

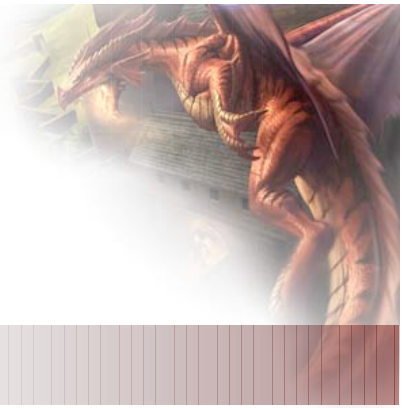
Fall 2005 *2005-09-27*

Version 1.0



Philosophical Aside - Software Arts

- Computing Science has more in common with making a movie, producing a play, writing a novel, and managing a company than it does with building bridges, making airplanes, and running railways.
- We should stop worrying about how to bring more engineering to CS and ask instead what can CS contribute and learn from the creative activities typical of the arts.

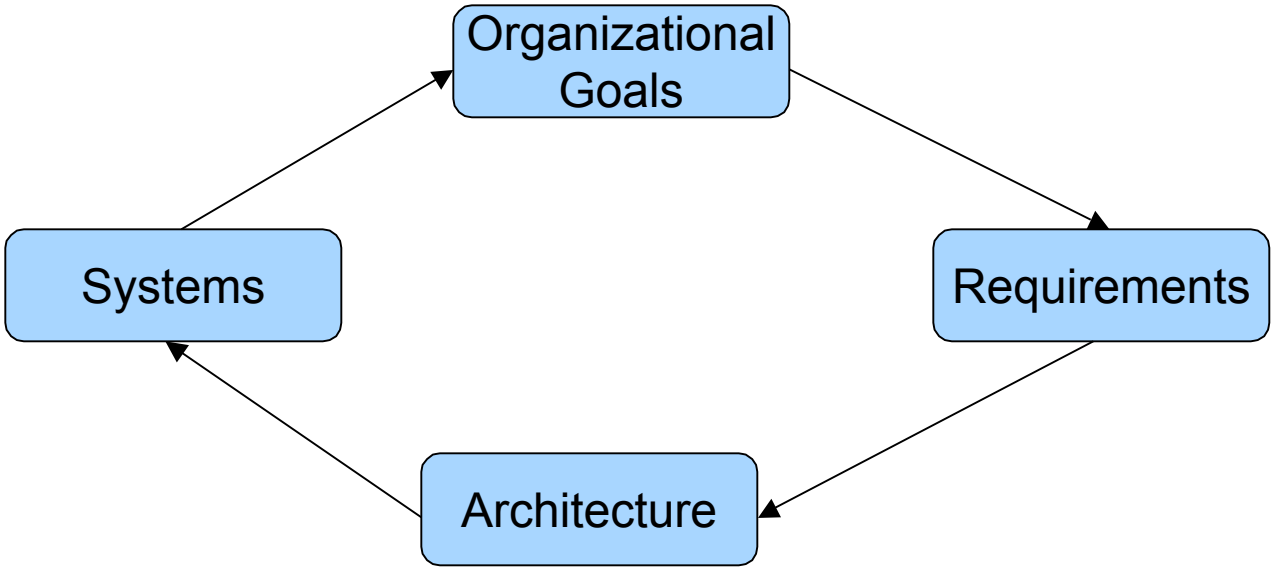


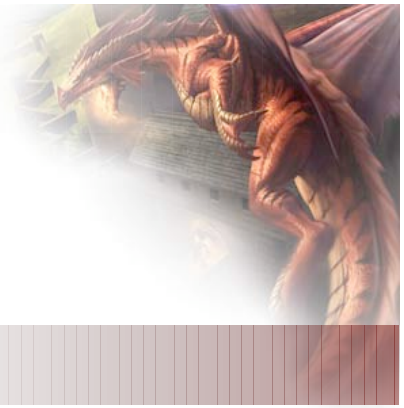
Activities vs Artifacts

- CS deals with the vague, rapidly changing, uncertain goals that characterize human activities, not the more stable properties typical of human artifacts.

Architecture Business Cycle

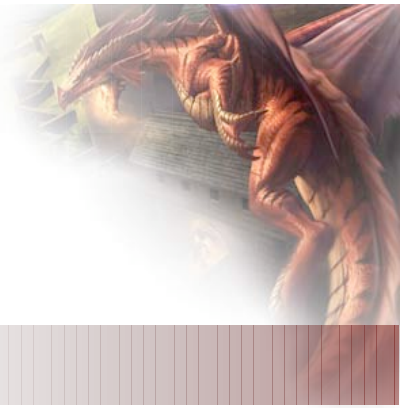
Len Bass





ABC

- ♣ Organizational goals influence requirements.
- ♣ Requirements lead to architecture.
- ♣ Architectures yield systems.
- ♣ Systems suggest evolution of organizational capabilities and requirements.

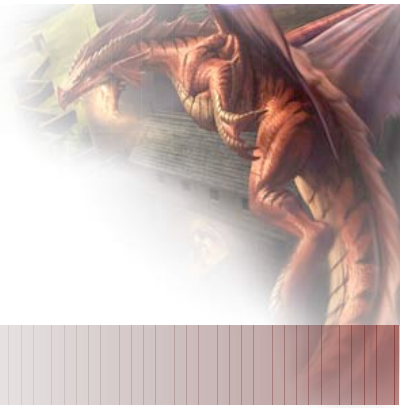


What is Architecture

(Webster Online)

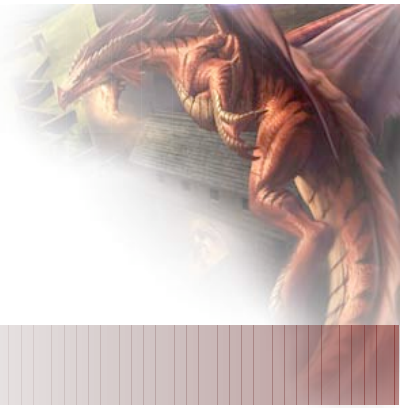
- The manner in which the components of a computer or computer system are organized and integrated.
- A unifying or coherent form or structure. <the novel lacks architecture>

CS architecture principles apply to any domain



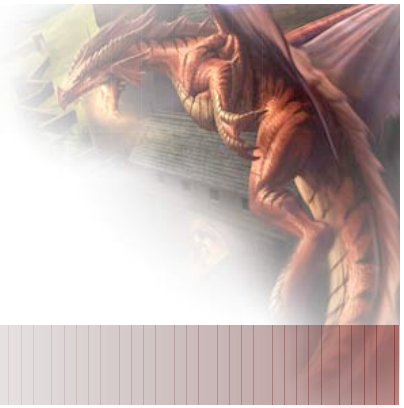
Iterative Process

- Applications must evolve
- At the same time maintaining architectural integrity.
- The architecture must take into account both current needs and future change.
- Good reference architectures can accommodate typical kinds of evolution since these changes follow well known paths.
- The challenge: handle new opportunities without causing too much brittleness in the architecture.



Iterative Process …

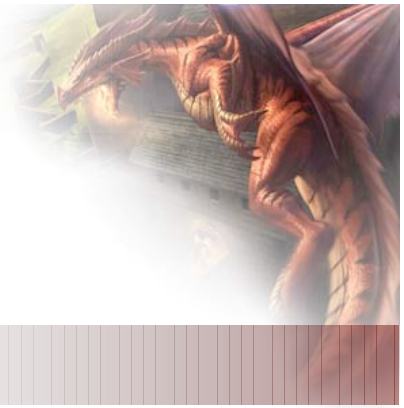
- An iterative and incremental approach has the following advantages:
- The application exists early and works early.
- It allows early concentration on uncertain or troublesome aspects.
- It lowers integration time and cost, and moves integration costs to front of development cycle where costs can't be ignored.



Iterative Process …

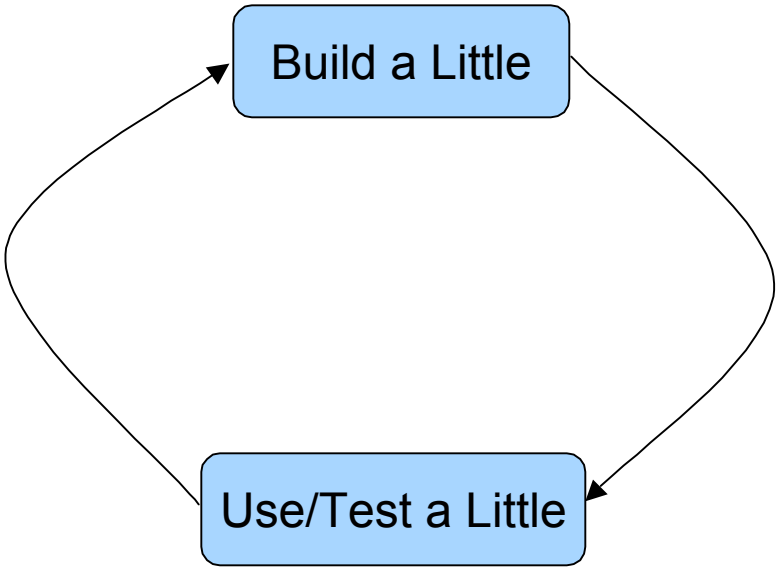
- Since the system always exists, testing and review can begin early, and performance and workflow issues surface early when cheaper to fix.

- Complex dependencies and interactions between modules are found earlier, and interface evolution to accommodate this occurs early.



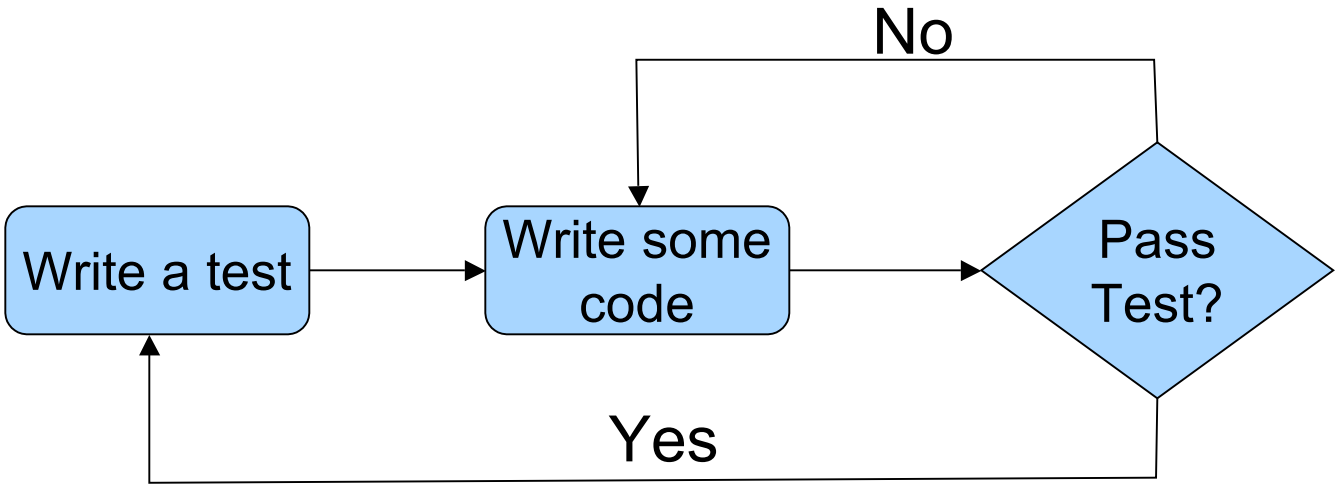
ABC implies that system building is iterative

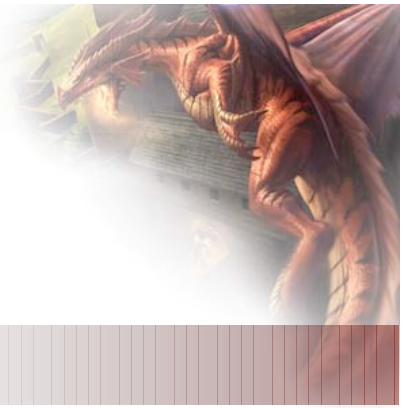
Requirements gathering



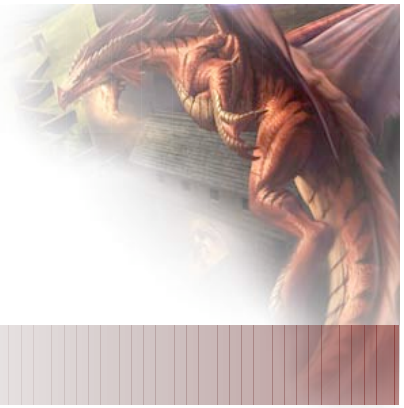
Agile Development

- Tests capture requirements.
- Build to the tests.



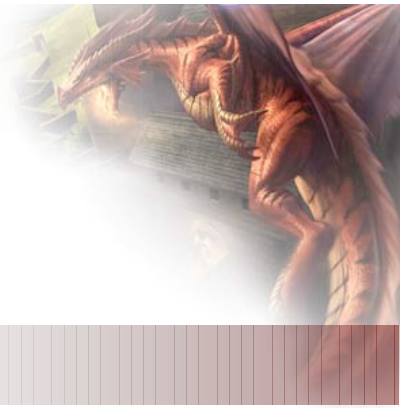


This only works if you can perform tests!



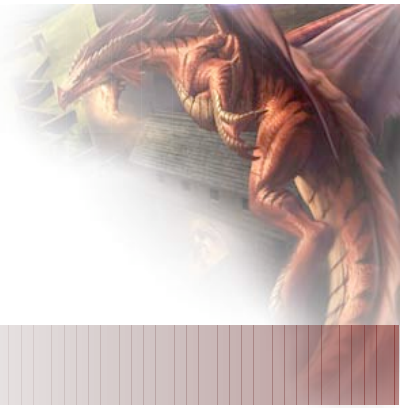
Tests? What tests?

- Some kinds of programs easily tested: given this input, what output do we expect
- GUI-based programs harder, need some kind of test script, and hand play.
- Ideas are tested by reviews and prototypes.



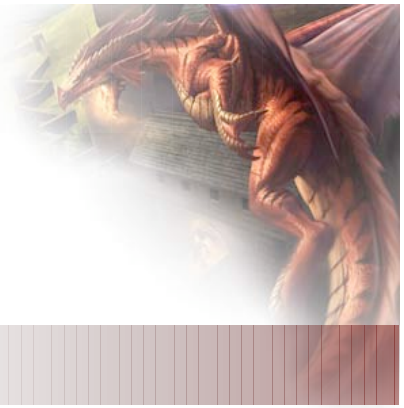
Scripted Tests

- Tester follows a script of steps of the form: do this, expect this and checks the results against the expected outputs (within error)
- At branch points checkpoint the world so you do not have to go back and replay
- Keep the scripts around as regression and acceptance tests



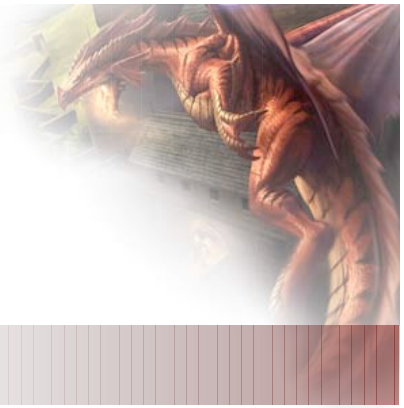
Kinds of Tests

- Regression: used to see if a change has broken something
- Acceptance: used to demonstrate that the system does roughly what was intended



Kinds of Tests

- Manual: a human follows the script and checks the results (tedious)
- Automated: a test harness executes the script and checks the results (repeatable)



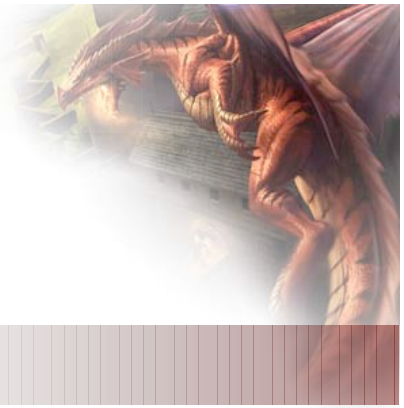
Test Design

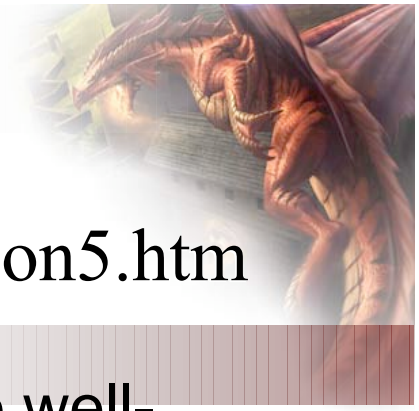
- When can you start to test?
- What can you test?
- How do you make testing easier?
- Can you design for testing?
- How do you test in the presence of multiple interacting modules?
- If a test fails, how do you isolate the problem?
- How much does a test designer need to know about the internal workings and design decisions?

Feature Interaction Problem

- As systems become more complex, components that work individually may fail when put into a cooperating group.
- Origin in the telecom industry
- What about computer games?

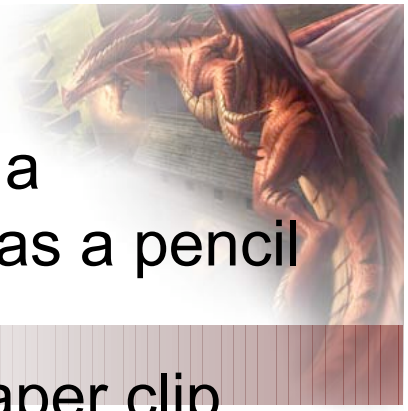
Testors



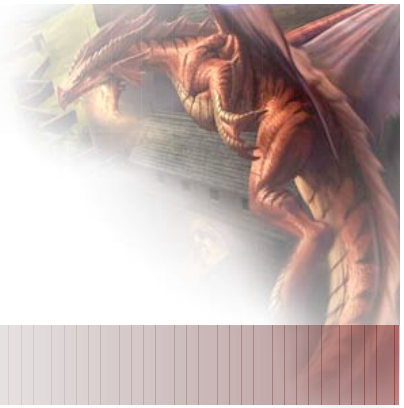


From http://www.sloperama.com/advice/lesson5.htm

"Verbal communication skills. The tester must be well-spoken. Words that come out of a tester's mouth must convey his thoughts clearly, giving information to the listener. Imagine these two exercises, which will help the tester in developing verbal communication skills. How a tester performs in these exercises also reveals the level of his existing verbal communication skills. Both of these exercises are best performed in neighboring cubicles -- the two people taking part in the exercise can easily converse but cannot see what the other is doing."



“The paperclip exercise. The tester must describe a randomly-bent paper clip to another person who has a pencil and paper. The goal is for the tester to get the listener/"customer" to draw a picture of the bent paper clip, without the tester ever saying the words "paper clip" or describing what the object is made of or was originally used for in any way whatsoever. Simply describing how the paper clip looks in its present state, the tester must obtain a correct picture of the paper clip on the second person's piece of paper. It can be enlightening for the tester to see what the drawing looks like, after completing the exercise. This exercise can also be performed using pipecleaners or twist-ties. The clip should be bent in a flat (2D) shape, not a 3D shape, since the listener/"customer" is drawing on 2D paper.”



"The building blocks exercise. This exercise is used at Nintendo of America to train or test their Customer Support representatives. Both parties to the exercise have identical boxes of wooden building blocks. The tester builds a structure from his building blocks and describes his structure to the other participant in the exercise. If the tester does it well, the two structures will be identical. If the two structures are not identical, the tester can learn how he ought to improve."