

Design and Culture

John Reekie
University of Technology, Sydney

Terms of Use: Creative Commons Attribution-ShareAlike 2.5
<http://creativecommons.org/licenses/by-sa/2.5/>

“Culture”?

- ◆ The predominating attitudes and behavior that characterize the functioning of a group or organization.

Let's have a surprise quiz...

<http://www.dictionary.com>

What do you most want?

- A. Money
- B. Power
- C. Success
- D. A new girl/boyfriend

What's your definition of success?

- A. A big house
- B. A beautiful wife/husband
- C. Sleeping in on Sunday morning
- D. Creating software that people use

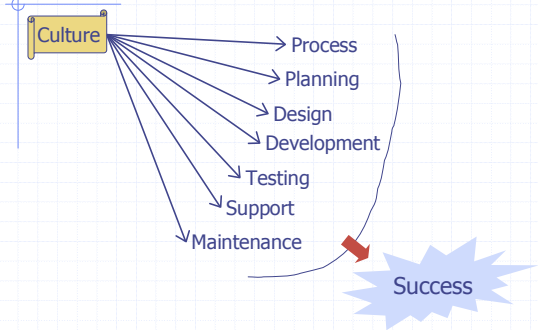
Will you be able to create software that people use?

- A. Yes
- B. No
- C. Maybe
- D. Maybe not

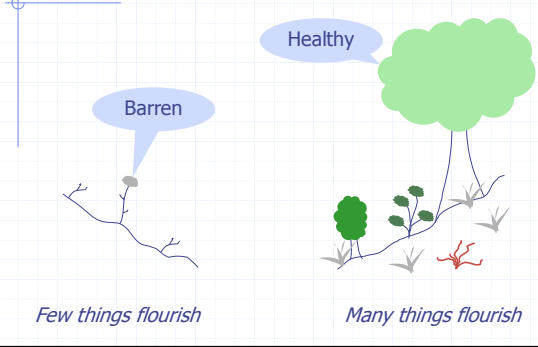
Why? (or why not?)

- ◆ No job
- ◆ No process
- ◆ No planning
- ◆ No architecture

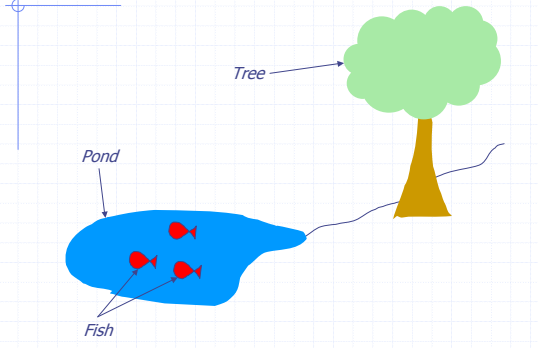
Or.... none of the above!

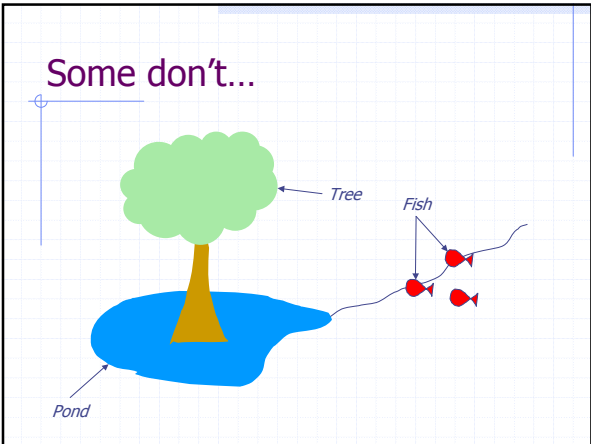


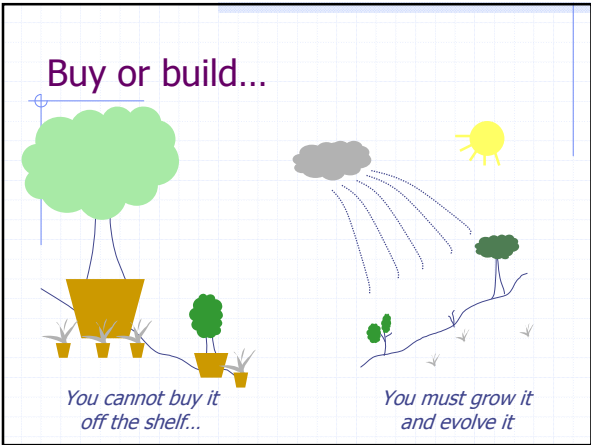
Culture – a bit like an ecosystem?



Some ecosystems work...









A healthy culture

- ◆ Strive to improve
- ◆ Peer review
- ◆ Personal growth

- ◆ Non-judgmental mentoring
- ◆ Teamwork by default
- ◆ Get it right the first time
- ◆ Clearly defined roles

- ◆ Collect metrics
- ◆ Ongoing education
- ◆ Clear strategies

The most widely-deployed architecture in existence is...

An illustration of many forces

The Big Ball of Mud

- ◆ Built from common, inexpensive materials and simple tools
- ◆ Little specialization of skills
- ◆ Little planning or regulation of growth
- ◆ Fulfill an immediate, local need

Brian Foote and Joseph Yoder
Big Ball of Mud
<http://www.laputan.org/mud/>

Sound familiar?



- ◆ Quick and dirty code takes on a life of its own
- ◆ Inadequate investment in tools and infrastructure
- ◆ Under-investment in libraries and frameworks
- ◆ Problems in one part of the system erode other parts
- ◆ Relentless onslaught of changing requirements

Forces? Consequences?

- ◆ Lack of:
 - Time
 - Skilled architects
 - Experience
- ◆ "Low-road" architecture
 - Adaptable
 - Flexible
 - Easy to modify



- ◆ By understanding the forces that lead to a Big Ball of Mud, we can devise strategies for improving system health over its lifetime, and avoid having to *Throw It Away*

Piecemeal Growth

- ◆ Master Plans are often too rigid
- ◆ Allow opportunities for local growth

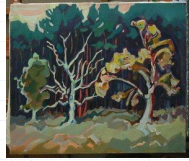


MIR space station

Keep it Working

- ◆ Once a system is working, don't allow changes that stop it from working

- Incremental development
- Nightly builds



The deep dark forest of non-working software

[Pavlenko Leonid, The Drevlyanskiy land](http://ukraine-art-gallery.com/)
<http://ukraine-art-gallery.com/>

Sweep it Under the Rug

- ◆ Cordon off messy architecture or code
- ◆ Provide interfaces to allow clean *new growth*



Chernobyl reactor #4

Parting thoughts...



Architecture astronauts

"When you go too far up, abstraction-wise, you run out of oxygen. Sometimes smart thinkers just don't know when to stop, and they create these absurd, all-encompassing, high-level pictures of the universe that are all good and fine, but don't actually mean anything at all.

"...Then they'll build applications like Groove that they think are *more* general than Napster, but which seem to have neglected that wee little feature that lets you type the name of a song and then listen to it -- the feature we wanted in the first place."

"Don't Let Architecture Astronauts Scare You"
Joel Spolsky
<http://www.joelonsoftware.com/articles/fog000000018.html>

Ivory tower architecture

You should beware ivory tower architectures. An ivory tower architecture is one that is often developed by an architect or architectural team in relative isolation to the day-to-day development activities of your project team(s). ... Ivory tower architectures are often beautiful things, usually well-documented with lots of fancy diagrams and wonderful vision statements proclaiming them to be your salvation. ...Ivory tower architectures are a significant risk to your project until you know they actually work through the concrete feedback provided by a technical prototype. ...[they] promote overbuilding of software because they typically reflect every feature ever required by any system that your architect(s) were ever involved with and not just the features that your system actually needs.

"Agile Architectural Modeling"
Scott Ambler
<http://www.agilemodeling.com/essays/agileArchitecture.htm>

Software has no shape

Software has no shape. There is no diagram, no drawing, no picture, that can capture as much of what's going on in a software system as a blueprint, subway map, or schematic can. If you're lucky, you might find part of your system that has a shape, just as the plumbing in a house has a shape. Just don't expect the shape of that aspect to be as similar to some `shape of your whole system' as the shape of a house's plumbing is to the shape of a house.

THEREFORE:

Don't try to `draw' your software system!

In particular, use [CaseTools](#) to capture some of your thoughts, to help force you to think about what you're analyzing or designing ... but not to `draw a picture of your software system.'

WikiWikiWeb
<http://www.c2.com/cgi/wiki?SoftwareHasNoShape>

The role of the software architect

"A simplistic view of the role is that architects create architectures, and their responsibilities encompass all that is involved in doing so. This would include articulating the architectural vision, conceptualizing and experimenting with alternative architectural approaches, creating models and component and interface specification documents, and validating the architecture against requirements and assumptions."

"However, any experienced architect knows that the role involves not just these technical activities, but others that are more political and strategic in nature on the one hand, and more like those of a consultant, on the other. A sound sense of [business and technical strategy](#) is required to envision the "right" architectural approach to the customer's problem set, given the business objectives of the architect's organization."

"The Role of the Software Architect"
Ruth Malan and Dana Bredemeyer
<http://www.bredemeyer.com/who.htm>

That's all, folks!



Thus concludes this subject!
