

# Architecture Business Cycle

## CSSE 574: Session 8, Part 2

Steve Chenoweth

Phone: Office (812) 877-8974, Cell (937) 657-3885

Email: [chenowet@rose-hulman.edu](mailto:chenowet@rose-hulman.edu)

# This discussion

- Start the general topics in Bass
  - Background for the specific skills you'll build
- How software architecture fits in with
  - Your company's business, and
  - The businesses of your customers
- Taken from Bass's Ch 1 - 4

# Outline

- Vasa – first of many case histories
  - We learn from others' mistakes!
  - Critical thinking's finest hour?
- Architecture Business Cycle
- Advice and Rules of Thumb for "Good" Architecture

Naval architecture – an interesting analogy to software architecture, as we'll see...

# History of the Vasa

- January 1625: Initial contract
- Early 1626: Accelerated schedule as building commenced
- Later 1626: Ship size increased
- 1627: Shipwright died
- August 1628: Maiden voyage



# Schedule Pressure

- Original project was to take 4 years
- Shorter schedule demanded by King →
- Ship builders opted to extend small ship rather than start over



Gustav II Adolf

# Changing Requirements

- King asked for larger size to accommodate more guns (Big war with Poland, not going well)
- Eventually asked for 2 decks of guns

# No Written Specifications

- Original ship was familiar to ship builders
- Scaling up was done without written specifications

# No Written Plan

- 3 "managers" worked without written plans
- Shipwright died during project
- 400 people working in 5 different groups: largest project in Sweden at this time

# Missing Science

- 17th century ship builders could only measure stability and heeling characteristics by trial
- Center of gravity could not be predicted accurately

# Cross-Section of Vasa



# Failed Stability Test

- Lurch test conducted by 30 men running back and forth on deck
- Test stopped when it became clear that ship would capsize
- Neither of the 2 remaining project managers attended the test

# Result - on Maiden Voyage :



# Stakeholders

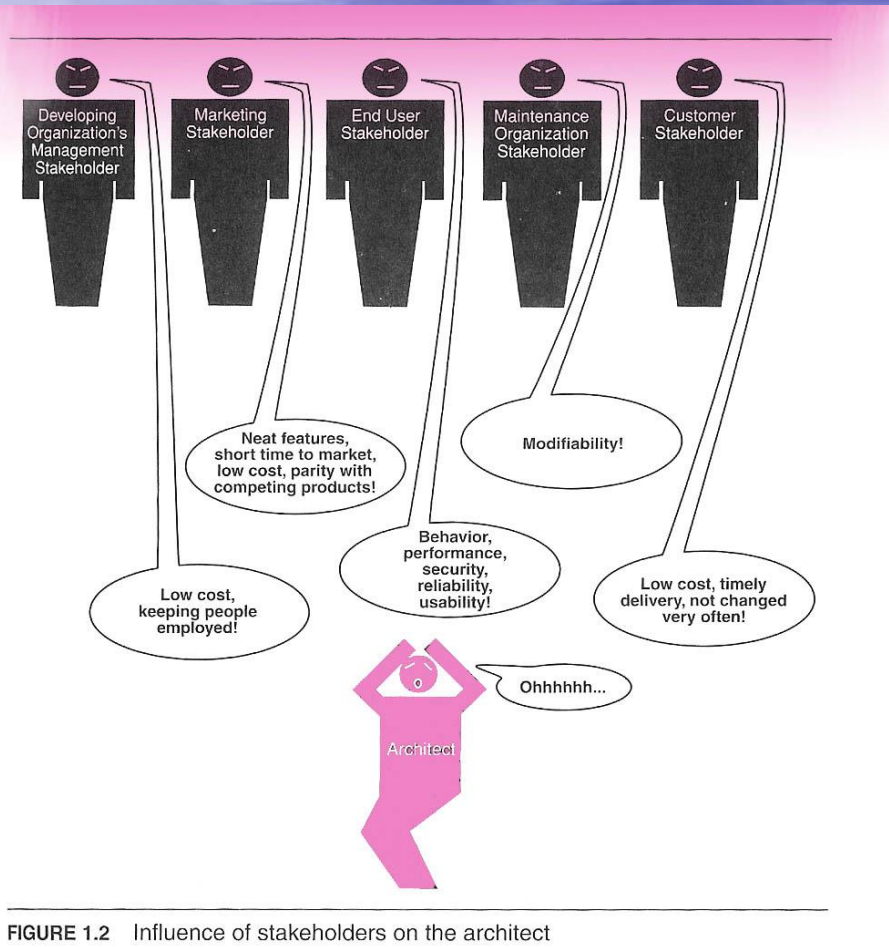
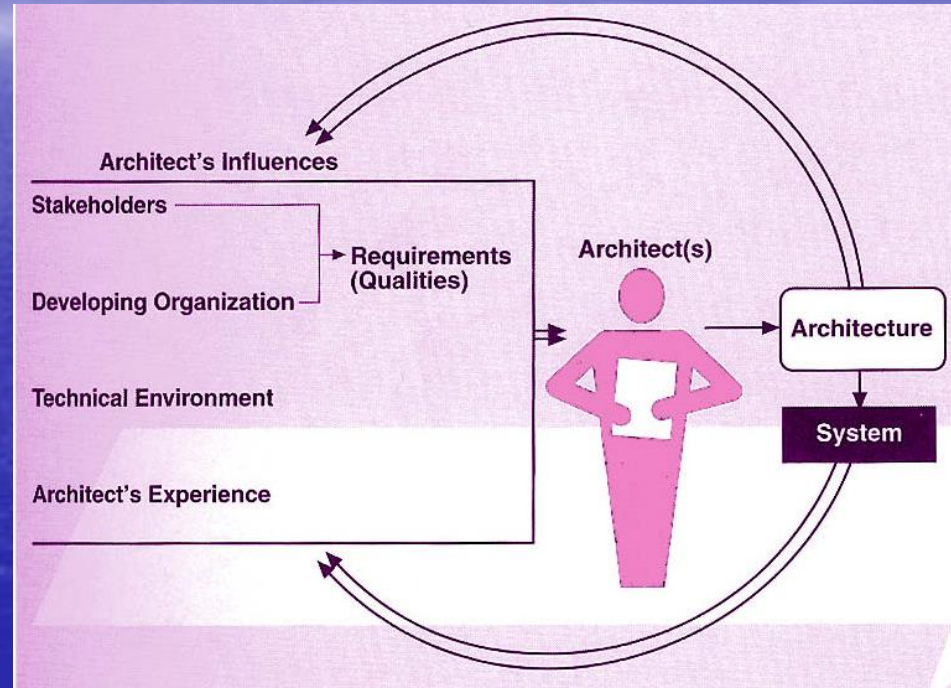


FIGURE 1.2 Influence of stakeholders on the architect

- Developing Organization Management
- Marketing
- End User
- Maintenance Organization
- Customer

# Architecture Business Cycle

- Stakeholders
- Developing Organization
- Technical Environment
- Architect's Experience



Steve's useful heuristic for "what it's like" – It's "technical leadership" :  
1/2 social activity, in the middle of *everyone* else, listening to & selling ideas  
1/2 creating great designs that lead to success

# Architecture Activities

- Creating business case
- Understanding requirements
- Creating or selecting architecture
- Documenting and communicating architecture
- Analyzing architecture
- Implementing system
- Ensuring conformance of implementation

# Architecture Process Advice (1/4)

1. Architecture should be product of a single architect or small group with identified leader

University of Oregon's TeachEngineering System Vision and Design team. From [test.teachengineering.org/founding\\_partners.php](http://test.teachengineering.org/founding_partners.php).



# Architecture Process Advice (2/4)

2. Architect should have functional requirements and a prioritized list of quality attributes
3. Architecture should be well-documented with at least one static and one dynamic view

↖ You've been doing this!

# Architecture Process Advice (3/4)

4. Architecture should be circulated to stakeholders, who are active in review
5. Architecture should be analyzed (quantitatively and qualitatively) before it is too late.

In the building trades, an architect also consults with the clients and users: Curtis J. Moody, FAIA, Columbus, OH architect, reviewing building plans with school principal Monica Grant. From web site [http://www.architectureweek.com/cgi-bin/awimage?dir=2001/0718&article=culture\\_1-2.html&image=11474\\_image\\_3.jpg](http://www.architectureweek.com/cgi-bin/awimage?dir=2001/0718&article=culture_1-2.html&image=11474_image_3.jpg).

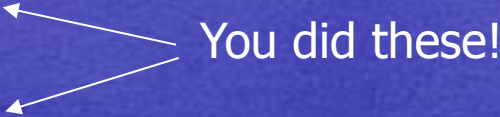


# Architecture Process Advice (4/4)

We'll continue doing this!

6. System should be developed incrementally from an initial skeleton that includes major communication paths
7. Architecture should result in a small number of specific resource contention areas

# "Good" Architecture Rules of Thumb (1/3)

1. Use information hiding to hide computing infrastructure
  2. Each module should protect its secrets with a good interface
  3. Use well-known architecture tactics to achieve quality attributes
- 
- You did these!

# "Good" Architecture Rules of Thumb (2/3)

4. Minimize and isolate dependence on a particular version of a commercial product or tool.
5. Separate producer modules from consumer modules.
6. For parallel-processing, use well-defined processes or tasks.

# "Good" Architecture Rules of Thumb (3/3)

7. Assignment of tasks or processes to processors should be easily changeable (even at runtime)
8. Use a small number of simple interaction patterns