AntiPatterns

DEVELOPMENTOR

DEVELOPING PEOPLE WHO DEVELOP SOFTWARE

What is an AntiPattern?

- Commonly occurring solution that has negative consequences
- Can be caused by
  - Insufficient experience of environment
  - Lack of knowledge
  - Right pattern, wrong context
- Documentation describes
  - How to recognize the AntiPattern
  - Consequences of the AntiPattern
  - How to move from the AntiPattern to a solution

AntiPatterns Areas

- AntiPatterns pop up in many areas of a project
  - Development
  - Architectural
  - Managerial

Development AntiPatterns

- Development AntiPatterns
  - Define classic problems in OO software
  - Should also specify how to refactor to a solution

The Blob

- One class takes on all the responsibilities
- Symptoms
  - Single class with lots of attributes/methods
  - Class with lots of unrelated attributes/methods
- Causes
  - Lack of understanding of OO design
  - Lack of architecture of architectural enforcement
  - Poor specification
- Solutions
  - Categorize related attributes/methods in contracts
  - Migrate code into related classes

Spaghetti Code

- Classic AntiPattern
- Symptoms
  - Code is difficult to maintain reuse
  - Lack of object orientation
  - Large complex methods
- Causes
  - Inexperience with OO design
  - No mentoring
  - No design up front
- Solutions
  - Software refactoring/code cleanup
  - (but call it “software investment” to appeal to managers!)

Lava Flow

- Code that probably serves no purpose
  - But cannot be removed “just in case”
  - Like blocks of basalt in a lava flow
- Symptoms
  - Unused code
  - Obsolete interfaces
- Causes
  - Prototype code moved to production
  - Uncontrolled distribution of research code
  - Single developer written code
  - Reworked application
- Solutions
  - Design first
  - Establish stable interfaces/contracts

Golden Hammer

- When all you have is a hammer
  - Everything looks like a nail
- Symptoms
  - Identical tools/frameworks used for disparate applications
  - System architecture best described by a particular product suite
- Causes
  - Large investment in technology/product
  - Isolation from industry
- Solutions
  - Philosophical – need to try new/other things

Copy and Paste Programming

- Similar segments of code scattered through the application
- Symptoms
  - Same bug in multiple places
  - Finding and fixing all instances of a problem is hard
- Causes
  - Laziness
- Solutions
  - Refactor into re-useable classes/libraries

Other AntiPatterns

- Continuous Obsolescence
  - Technology that changes so rapidly that it's hard to keep pace
  - DDE, OLE, COM, ActiveX, DCOM, COM+
  - .Net 1.0, 1.1, 2.0, 3.0, 3.5, …
- Mushroom Management
  - Keep them in the dark and feed them …
- Boat Anchor
  - Costly acquisition that must be used
- Input Kludge
  - End users can break new programs too easily

Architecture AntiPatterns

- System and Enterprise level problems
  - Common mistakes in the creation, implementation and management of architecture

Stovepipe Enterprise

- Stovepipes are typically patched up
- Symptoms
  - Incompatible technologies in the enterprise
  - Brittle system architectures
  - Lack of interoperability
- Causes
  - Lack of enterprise level strategy
  - No incentive for co-operation
  - Lack of technological knowledge
- Solutions
  - Co-ordinate technology across the enterprise
  - Agree on standards within the enterprise

Stovepipe System

- Single system analogue to the Stovepipe Enterprise
  - But across a set of applications
- Symptoms
  - Documentation and application do not match
  - Requirement changes costly to implement
  - System maintenance costs high
- Causes
  - Lack of architecture
  - Lack of abstraction
- Solutions
  - Use contract based component architecture

Architecture by Implication

- Architecture, what architecture
- Symptoms
  - Lack of architecture planning and specification
  - Lack of knowledge of domain, scale, technology, security persistence, etc.
- Causes
  - No risk management
  - Overconfidence of mangers/architects/developers
  - Reliance on previous (invalid) experience
- Solutions
  - Have an organized approach to systems architecture definition

Design By Committee

- Horse with a hump anyone?
- Symptoms
  - Overly complex design document
  - No decisions reached in meetings
  - Politically charged work environment
- Causes
  - No single project architect
  - Trying to keep everybody happy
  - Meetings are too large
  - No software process
- Solutions
  - Reform the meeting process
  - Use meeting facilitators
  - Have a single technical lead

Other AntiPatterns

- Auto-generated Stovepipe
  - Migrate existing software to a distributed system using the same design
- Cover Your Assets
  - Documentation is not specific
  - Too many alternatives discussed (this provides cover for the authors)
- Warm Bodies
  - Developers who don't deliver/Teams that are too large
- Swiss Army Knife
  - Excessively complex solution

Management AntiPatterns

- Describes how projects are impaired by
  - People issues
  - Resources
  - Processes
  - External relationships

Analysis Paralysis

- Symptoms
  - Multiple project restarts and model re-work
  - Design/implementation issues re-introduced during analysis phase
  - Analysis model is extremely complicated
- Causes
  - Waterfall models don't work
  - Management confident in analysis but not in development
  - 'All analysis must be finished first'
- Solutions
  - Incremental development

Death by Planning

- Symptoms
  - Lack of pragmatic planning
  - Focus on costs
  - Ignorance of projects actual status
  - Project overruns
- Causes
  - No up-to date plan
  - Poor project management skills
- Solutions
  - Project plan must show deliverables
  - Plan must be updated frequently (at least weekly)

Irrational Management

- Symptoms
  - Decisions not made
  - Wrong priorities
- Causes
  - Lack of management skills
  - Inability to make decisions
- Solutions
  - Admit the problem and seek help!
  - Provide short term goals
  - Share a focus

Other Patterns

- Blowhard Jamboree
  - So called industry experts exert too much influence
- Intellectual Violence
  - Knowledge used to intimidate others
  - Not always deliberate
- Smoke and Mirrors
  - Demonstration systems that promise too much
- Throw it over the Wall
  - Code finished but no documentation or testing!
- Fire Drill
  - Delays lead to short development times
- E-mail is Dangerous
  - Don’t use email to argue!

Summary

- AntiPatterns fall into three broad categories:
  - Development
  - Architecture
  - Management
- Be aware of these issues and solutions
  - Will let you catch problems up front