Objectives

- Bash scripting
- Software Tools
- Eclipse

Review

- How do we create a function?
  - How do we access its parameters?
- How do we call a function?
- How can we write a case statement?
  - What special feature do the conditions have?

Compound Command [[ ]]

- Expressions are built-in with the `[[ ]]` operator
  `if [[ $var = "" ]] …`
- Gets around *parsing* quirks of `/bin/test`, allows checking strings against *patterns*
- Operations:
  - *string* `==` *pattern*
  - *string* `!=` *pattern*
  - *string1* `<` *string2*   so don't need to escape <
  - *file1* `–nt` *file2*   is file1 newer than file2?
  - *file1* `–ot` *file2*   is file1 older than file2?
  - *file1* `–ef` *file2*   is file1 a hard link to file2?
  - `&&, ||`

smalltest.sh
arrays.sh

Compound Command: Patterns

- Can be used to do string matching:
  `if [[ $foo = *a* ]]`
  `if [[ $foo = [abc]* ]]`
- Similar to regular expressions, but different syntax

Additonal Parameter Expansion

- ${#*param*} – Length of *param*
- ${*param#pattern*} – Left strip min *pattern*
- ${*param##pattern*} – Left strip max *pattern*
- ${*param%pattern*} – Right strip min *pattern*
- ${*param%%pattern*} – Right strip max *pattern*
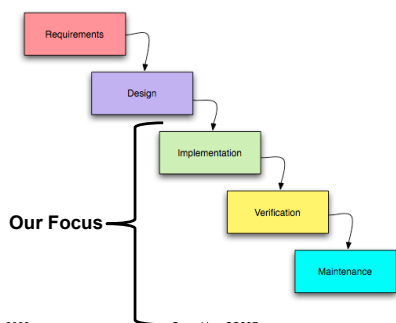- ${*param–value*} – Default *value* if *param* not set

Additional Features

- Built-in arithmetic: Using $((*expression*))
  - e.g., `print $(( 1 + 1 * 8 / x ))`
- Tilde file expansion

`~`	$HOME
`~user`	home directory of user
`~+`	$PWD
`~-`	$OLDPWD

1

Software Life Cycle: Waterfall Model

Requirements

Design

Implementation

Verification

Our Focus

Maintenance

Iterative Design

Requirements

Design

Test/
Get feedback
from users

Evaluate   Implement

Deployment
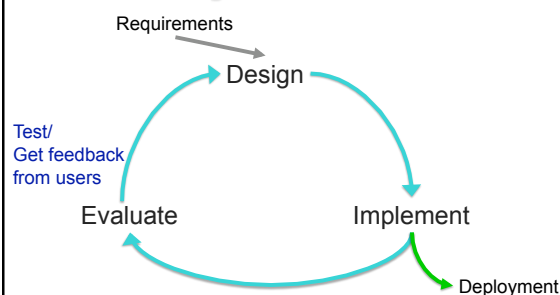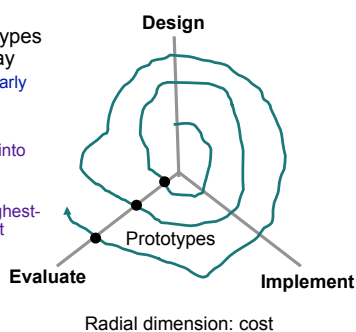
Spiral Model

- Idea: smaller prototypes to test/fix/throw away
  - ➢ Finding problems early costs less
- In general…
  - ➢ Break functionality into smaller pieces
  - ➢ Implement most depended-on or highest-priority features first

Design

Evaluate

Prototypes

Implement

Radial dimension: cost

Goal: Productivity

- Speed up common tasks and difficult tasks

- What is common/hard about software development?

Common Tasks

- Collecting/compiling source
  - ➢ Libraries (3rd party or otherwise)
- Package/distribute code
- Finding code
  - ➢ Debugging, maintenance tasks
- Documenting code
- Redesigning code
  - ➢ For performance, extensibility, reusability, …
  - ➢ "refactoring"
- Profiling

- Debugging
  - ➢ Find source
  - ➢ Fix source → won't cause new bugs?
- Keeping track of bugs
  - ➢ Bugs found, symptoms
  - ➢ Which bugs fixed
- Testing
  - ➢ Thorough tests
  - ➢ Rerun after code changes
- Collaborating between developers
  - ➢ Multiple users

Explore tools to help
with these tasks

Demos

- In 2s or 3s, present 3 tools
  - ➢ Assigned group
  - ➢ Working on schedule
- Learn tool
  - ➢ Motivation, background
  - ➢ Download, install, use
  - ➢ Pros and cons of tool, strengths, limitations
    - What you like, what you don't
- Lead us through installation, an example use
- I'll do the first few

Eclipse Platform

- Projects focus on building an open development platform comprised of extensible frameworks, tools and runtimes for *building, deploying and managing software across the lifecycle*
- Open source
- Originally started by IBM in 2001
- Most commonly used for Java development
  - Enterprise Java applications, e.g., web
- Other IDEs (Sun's NetBeans, Microsoft Visual Studio)
  - Has marketshare for Java
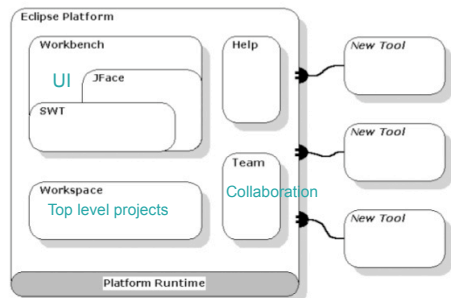
May 4, 2009 — Sprenkle - CS297

Eclipse Platform Plugins

- Eclipse has a lot of built-in functionality
  - Most of functionality is generic
  - Need additional tools to extend Eclipse to work with new content types (e.g., HTML rather than Java), to do new things with existing content types, and to focus generic functionality on something specific
- **Plugins:** Modules/tools
  - Many plugins available www.eclipseplugincentral.com
  - Integrate with other tools also written for Eclipse platform

May 4, 2009 — Sprenkle - CS297

Eclipse Platform

Source: Eclipse Platform Technical Overview

May 4, 2009 — Sprenkle - CS297

Using Eclipse

- Eclipse is installed on the lab machines
  - To find out where it's installed run
    - `which eclipse`
  - To install on your own machine, download the version: Eclipse IDE for Java EE Developers
    - `http://www.eclipse.org/downloads/`

- In a terminal, start Eclipse
  - `eclipse &`

May 4, 2009 — Sprenkle - CS297

Eclipse Tutorial

May 4, 2009 — Sprenkle - CS297

For Wednesday

- Assignment 6
- Read and summarize/analyze "Source Code Exploration with Google"
  - Login to Sakai with username and password used for WebAdvisor
  - Write summary on Forum
    - Post reply to appropriate topic
  - Answer analysis questions, e.g., how would you use the tool?
  - **Due** by 9 a.m. Wednesday

May 4, 2009 — Sprenkle - CS297