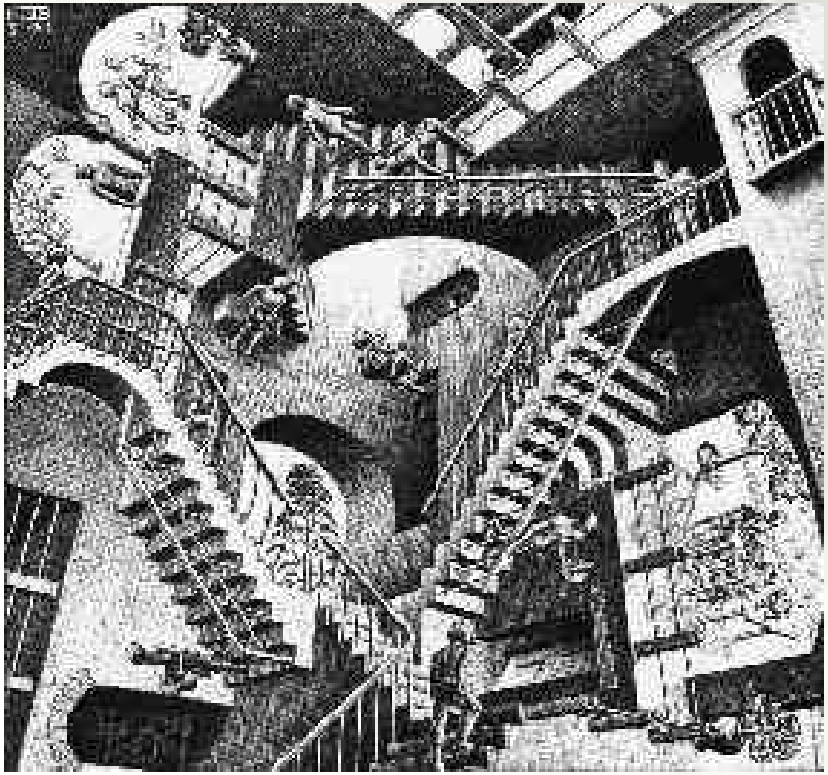
# Continuous Integration

Continuous Integration using Damage Control

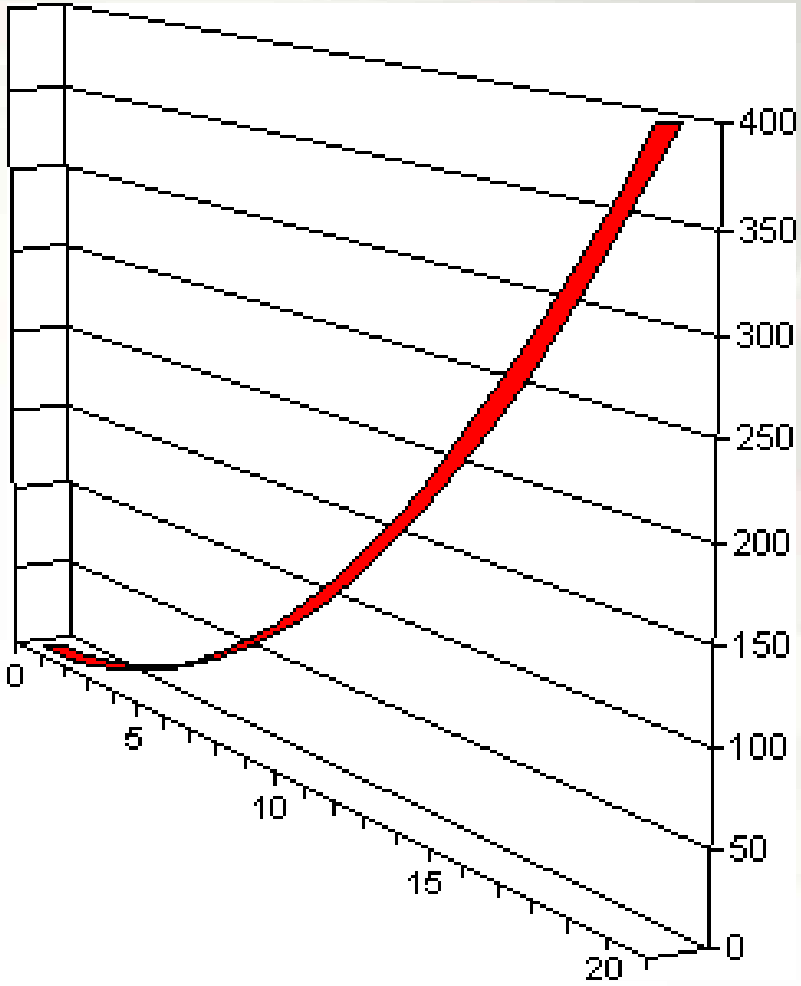
Avik Sengupta

# Integration



- Making large, disparate parts working together
- Traditionally, error prone, ... and dreaded
- Solution – Integrate often
- Need tools to make it successful

# The Exponential Cost of Change




- The earlier you find an error, the easier it is to fix
- Empirically 10x cost escalation at each step from requirement, to developement, to QA, to release

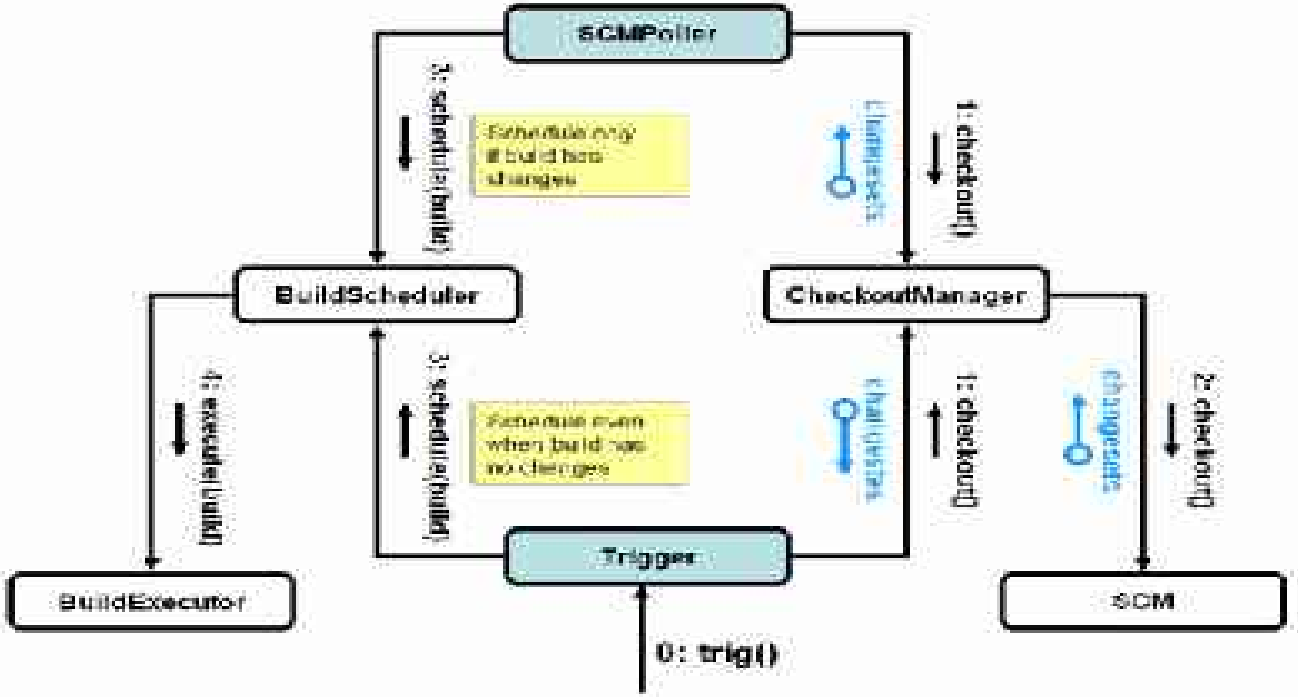
# Continous Integration

- Merge
- Build
- Test
- Rinse 'n Repeat

# Damage Control

- Ruby based automated build server
- Source control integration to trigger builds
- Web based reporting and control
- Depends on having single click build and test environments for your projects
- Build tool agnostic

# Damage Control Architecture



# More Features

- Scheduled Builds
- SCM Polling
- Issue Tracker Integration
- Build artifact archiving
- ViewCVS Integration

# DEMO

## Using DamageControl

# Caveats

- Young and evolving project
- Documentation could be improved
- Ongoing changes ... new version out soon

# Alternatives

- CruiseControl
  - Build ant based projects
- CruiseControl.NET
  - Build .NET (nAnt) based projects
- GUMP
  - Global continuous integration

# Questions?

# More Info

- Main DamageControl Site
  - http://damagecontrol.codehaus.org/
- Martin  Fowler on Continuous Integration
  - http://martinfowler.com/articles/continuousIntegration.html
- Cost of Change
  - http://www.xprogramming.com/xpmag/cost_of_change.htm
- Cruise Control
  - http://cruisecontrol.sourceforge.net/
- This Presentation
  - http://www.sengupta.net/talks/

# Dependencies & Installation

- Install  Ruby
- Download binary
  - http://damagecontrol.codehaus.org/
- Install DC server
- Install DC SCM plugin