

Agile, Secure Cloud Application Development & Management

Adam Getchell

acgetchell@ucdavis.edu

<https://github.com/acgetchell>

<http://www.linkedin.com/in/adamgetchell>

ucdotnet@ucdavis.edu

UC Computing Systems Conference

August 7-9, 2011

University of California Merced

How to make Software

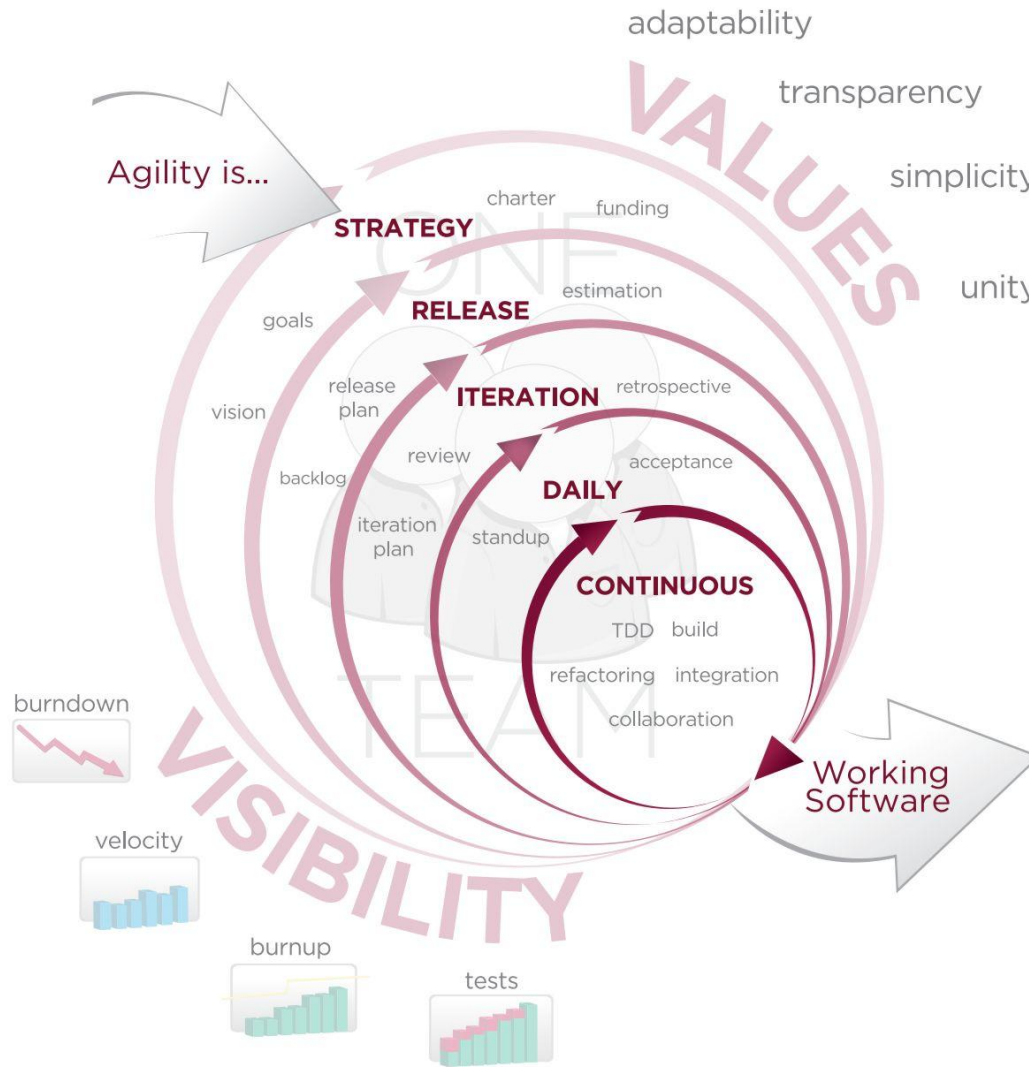
- Better*
- Faster
- Cheaper

* We posit that better = less bugs = greater security

Agile Development

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

AGILE DEVELOPMENT



ACCELERATE DELIVERY

Agile Development Principles

- Simplicity
- Self-organizing teams
- Regular adaption to changing circumstances
- Projects are built around motivated, trusted individuals
- Sustainable development, able to maintain a constant pace
- Welcome changing requirements, even late in development
- Continuous attention to technical excellence and good design
- Working software is the principal measure of progress
- Customer satisfaction by rapid delivery of useful software
- Close co-operation between business people and developers
- Face-to-face conversations as the best form of communication
- Working software is delivered frequently (weeks rather than months)

Agile Development Principles

- Simplicity
- Self-organizing teams
- Regular adaption to changing circumstances
- Projects are built around motivated, trusted individuals
- Sustainable development, at a constant pace
- Welcome changing requirements, even late in development
- Continuous attention to technical excellence and good design
- Working software is the principal measure of progress
- Customer satisfaction by rapid delivery of useful software
- Close co-operation between business people and developers
- Face-to-face conversations as the best form of communication
- Working software is delivered frequently (weeks rather than months)

Processes

Agile Development Principles

- Simplicity
- Self-organizing teams
- Regular adaption to changing circumstances
- Projects are built around motivated, trusted individuals
- Sustainable development, able to maintain a constant pace
- Welcome changing requirements, even late in development
- Continuous attention to technical excellence and good design
- Working software is the principal measure of progress
- Customer satisfaction by rapid delivery of useful software
- Close co-operation between business and developers
- Face-to-face conversations as the best form of communication
- Working software is delivered frequently (weeks rather than months)

Processes

Feedback



Cloud Computing

Having secure access to all your applications and data from any network device

Agile Development in the Cloud

Agile and the Cloud: Perfect Together*

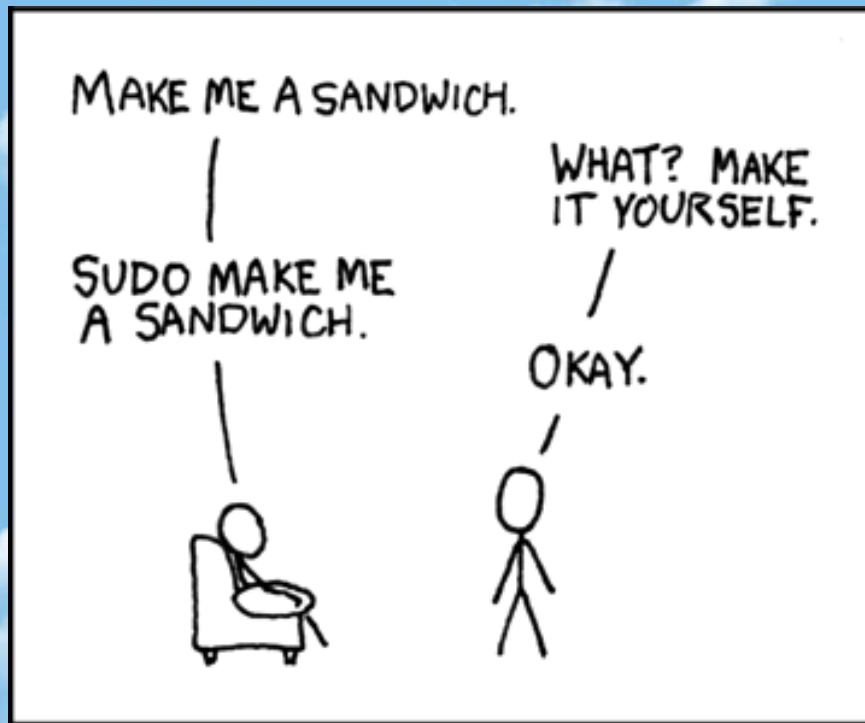
- Agile software: grant users of applications new capabilities as they need them
- Cloud computing: Pervasive, on-demand computation as a utility (elastic provision, pay-as-you-go, online, illusion of infinite supply)

*Jim Ensell, “Agile Development in the Cloud”,
<http://www.agilejournal.com/articles/columns/column-articles/6018>

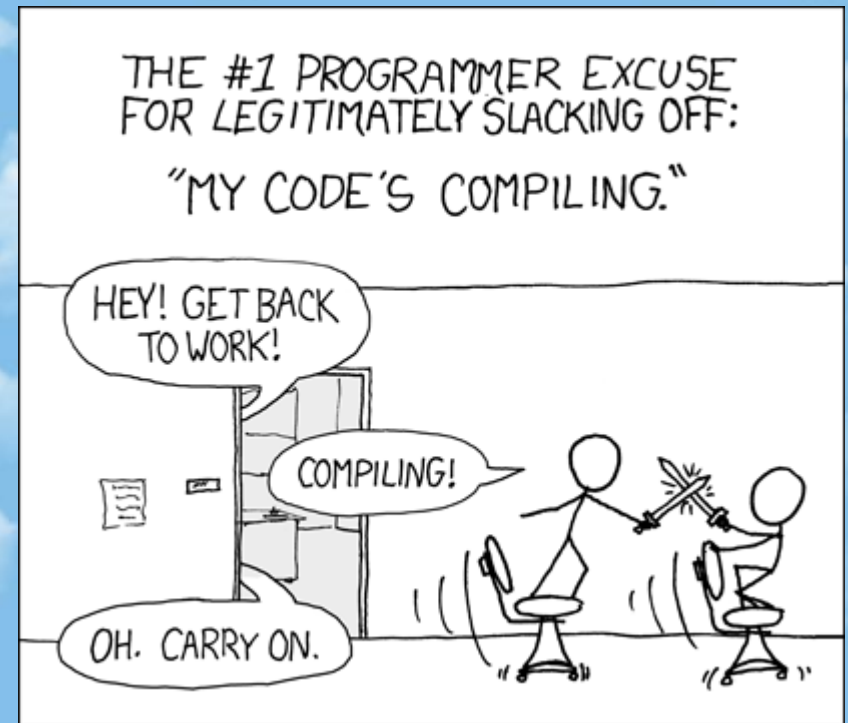
Stages of Agile Cloud Development

1. Developing using cloud-based source control management (SCM) repositories
2. Build in the cloud using virtual images, provide feedback to continuous integration (CI) servers
3. Testing in the cloud by building test machines
4. Production in the cloud, automated deployment, load-balancing, analytics

Programmer (noun): An organism that can turn caffeine and alcohol into code*



<http://xkcd.com/149/>



<http://xkcd.com/303/>

* <http://uncyclopedia.wikia.com/wiki/Programmer>

How to save caffeine by-products?

Source Control! But ...

- It should be available, always
- It should be secure
- It should checksum files and keep readable version histories
- It should be easy to manage projects, collaborate with folks, setup teams/perms
- It should be ~~easy to use~~ able to do hard things like fork/branch/merge
- It should allow for distributed development

Git

- Cross Platform/IDE friendly: Integrates into Visual Studio, Eclipse, and Xcode 4.x
- Distributed Development: Each developer gets a local copy of the entire development history, and changes are copied from one repo to another. Git protocol is fast/efficient and SSH friendly
- Non-linear development: Rapid, convenient branching and merging
- Efficient handling of large projects: Very fast, scales well, typically from one to several orders of magnitude faster than other version control systems. Efficient packed format for long-term revision storage.
- Cryptographic authentication of history: History is stored such that a particular revision/commit depends upon the complete development history leading up to that commit. It's therefore not possible to change old versions without being noticed. Tags can be cryptographically signed.
- Toolkit: Git is a collection of many small tools plus scripts as convenient wrappers. Easily used by humans and automation.

MINGW32:/c/Projects/Purchasing

add_vendor.aspx	home.aspx.vb	order_reuse_test.aspx
add_vendor.aspx.vb	hr_emailer.aspx	order_reuse_test.aspx.vb
admin_approve.aspx	hr_emailer.aspx.vb	process.aspx
admin_approve.aspx.vb	images	process.aspx.vb
approve.aspx	lab_history.aspx	receipt.aspx
approve.aspx.vb	lab_history.aspx.vb	receipt.aspx.vb
approvers.aspx	lab_orders.aspx	receive.aspx
approvers.aspx.vb	lab_orders.aspx.vb	receive.aspx.vb
budget.aspx	login.aspx	style
budget.aspx.vb	login.aspx.vb	test.aspx
budget_setup.aspx	members.aspx	test.aspx.vb
budget_setup.aspx.vb	members.aspx.vb	vendors.aspx
building_list.aspx	no_access.aspx	vendors.aspx.vb
building_list.aspx.vb	no_access.aspx.vb	view.aspx
ddlElements.xml	open	view.aspx.vb
dev_folder	ops.master	view_orders.aspx
edit_orders.aspx	ops.master.vb	view_orders.aspx.vb
edit_orders.aspx.vb	ops_viewstate.master	web.config

getchell@HAPKID0 /c/Projects/Purchasing <plantsciences>

\$ git checkout JMIE

Switched to branch 'JMIE'

getchell@HAPKID0 /c/Projects/Purchasing <JMIE>

\$ ls

admin default.aspx orders register.aspx.cs

blocks default.aspx.cs register.aspx

getchell@HAPKID0 /c/Projects/Purchasing <JMIE>

\$



acgetchell

Dashboard

Inbox 0

Account Settings

Log Out

Explore GitHub

Gist

Blog

Help



Search...



ucdavis



News Feed

News Feed

Pull Requests

Teams

Organization Settings

Switch Context ▾



jSylvestre pushed to master at ucdavis/NuSurvey about 5 hours ago



ba733c7 QuestionController Tests



jSylvestre pushed to master at ucdavis/NuSurvey about 7 hours ago



1351b27 WIP QuestionController edit tests



jSylvestre pushed to master at ucdavis/NuSurvey about 9 hours ago



622992a WIP QuestionController edit tests



jSylvestre pushed to master at ucdavis/NuSurvey 1 day ago



8cac91c WIP QuestionController edit tests



f040c52 Added check so that when editing a question, the related category...



3f7510a Added a check when editing a question that the passed survey id m...

[View comparison for these 3 commits »](#)

jSylvestre pushed to master at ucdavis/NuSurvey 1 day ago



ad0b435 QuestionController Create tests



8a37b3e WIP QuestionController Create tests

[View comparison for these 2 commits »](#)

jSylvestre pushed to master at ucdavis/NuSurvey 4 days ago



ea40edf WIP QuestionController tests

Repositories (20)

New Repository

Find a repository...

All Repositories

Public

Private

Sources

Forks

ucdavis/NuSurvey

ucdavis/EligibilityList

ucdavis/ESRA

ucdavis/Agribusiness

ucdavis/UCDArchTemplates

ucdavis/UCDArch

ucdavis/CDT

ucdavis/Commencement

ucdavis/Recruitments

ucdavis/metrics-net

[Show 10 more repositories...](#)

GitHub@UCDavis

Organization: <https://github.com/ucdavis>

- Public and private repositories
- Anyone with a free GitHub account can become a team member
- We currently have 20 Private and unlimited Public repositories that allow for unlimited collaborators (team members)
- \$37.50/month (25% educational discount) for 10 Private repos

Repository:

- A location storing a snapshot of a particular project, plus commit history and subsequent changes.

GitHub processes

Fork + Pull: Contributors fork the code (clone) and work on their own repository, then submit pull requests

- Unlike previous version control systems, every repository can be a master ← by convention only
- upstream, origin, master

Push: Agree upon a master repository, then push changes to it

- Use branch/merge to minimize conflicts
- origin, master

Some Caveats

Don't store sensitive configuration info (LDAP/API keys, database passwords, etc.)

Scott Kirkland's method (.NET)

1. All sensitive information should be stored in the <appSettings> section.
2. Create a file in your website root called AppSettings.config.

AppSettings.config:

```
<appSettings>  
  <add key="WOPRpassword" value="Joshua"/>  
  <add key="MapsKey" value="1234567890-AA"/>  
</appSettings>
```

Some Caveats

In your web.config, add the following property to your <appSettings> element:
file="AppSettings.config"

Web.config:

```
<appSettings file="AppSettings.config">  
    ...  
</appSettings>
```

3. At this point, the appSettings values in your web.config will be overwritten by those in AppSettings.config. Remove any sensitive information from the web.config file now, and you can also remove non-sensitive settings from AppSettings.config if desired (optional) because anything in AppSettings.config will overwrite matching values (or non-existent values) in your web.config section.
4. The last step is to make sure that your new secret AppSettings.config file is not checked into source control, and for that you can just add **AppSettings.config** to your **.gitignore** file. Also, please think of the children and be sure to copy your AppSettings.config file to the project share so others can find it (ex: W:\Devel\Projects\Recruitment).

Some Caveats

If you accidentally committed sensitive info to a repo:

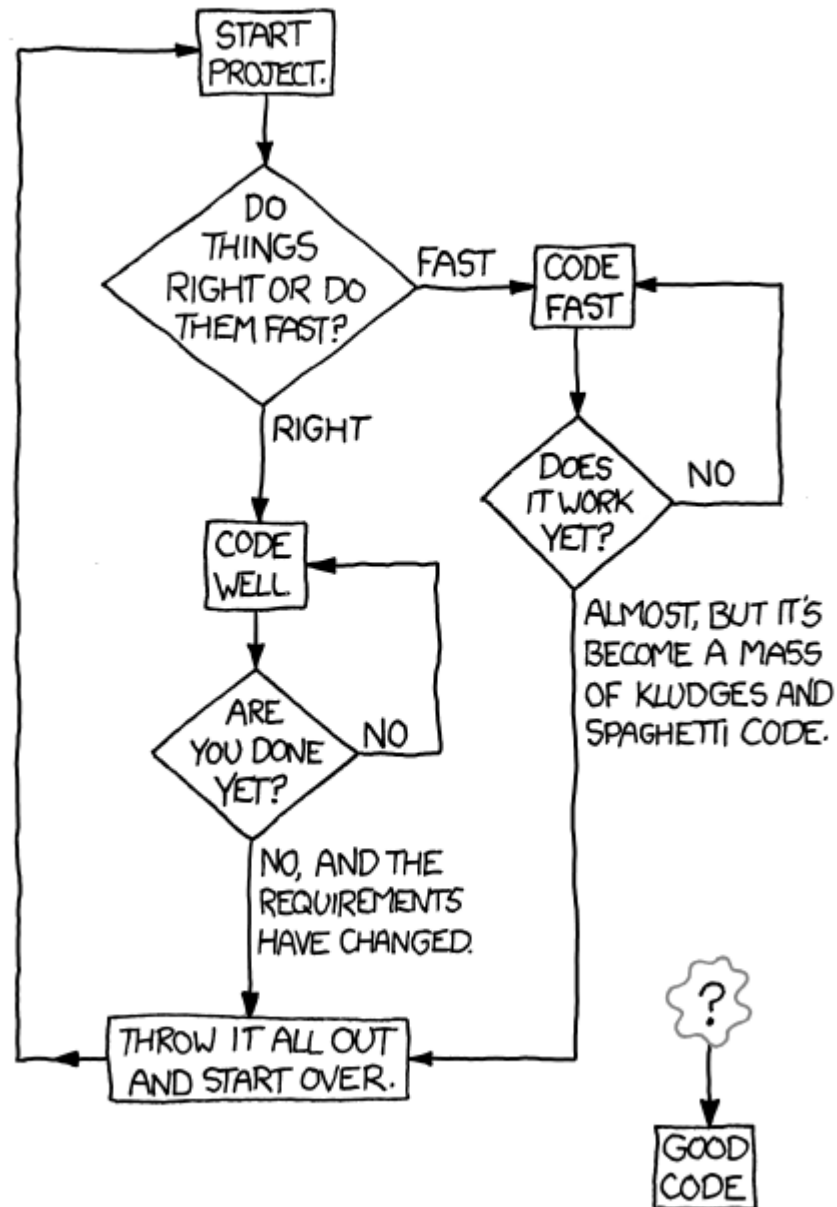
1. Change the password
2. Purge file containing sensitive info from repo
3. Cleanup and reclaim space
4. Tell collaborators to use *git rebase*

Full details here: <http://help.github.com/remove-sensitive-data/>

What have we accomplished?

- Simplicity
- Self-organizing teams
- Regular adaption to changing circumstances
- Projects are built around motivated, trusted individuals
- Sustainable development, able to maintain a constant pace
- Welcome changing requirements, even late in development
- Continuous attention to technical excellence and good design

HOW TO WRITE GOOD CODE:



<http://xkcd.com/844/>

Great! Now, what were we developing?

Code is logical

Business processes are ...

- Anyways, Agile development is nice and all, but how do we tell when we are done?
- And manage more than 1 project?
- And cope with changing requirements?*
- And user feedback?

*We don't. We make a contract with our users. And we never, ever, require them to look at GitHub to figure out how we're doing



UCDAVIS COLLEGE OF AGRICULTURAL AND ENVIRONMENTAL SCIENCES

[Login](#)

Dogbert Project Management

Welcome to the Dogbert Project Management System.

Use Dogbert to manage project priorities, requirements engineering and manage your IT projects. This tool will help managers prioritize their workgroup's projects and allows aids in managing projects that span workgroups. Dogbert will help staff manage projects that they are working on and provide easy documentation on projects. It is open to all UC Davis employees with a valid Kerberos account.

Dogbert has been used extensively on software projects for requirements engineering, use case tracking and document generation by the College of Agricultural & Environmental Sciences Dean's Office Computing Resources Unit.

To get started please sign up to request access. We will setup your account and a workgroup for you.

[Sign-Up](#)[Login](#)[Public Projects](#)



UC DAVIS COLLEGE OF AGRICULTURAL AND ENVIRONMENTAL SCIENCES

[Home](#) [Projects](#) [Tasks](#) [Admin](#)

WELCOME BACK ADAM

[Logout](#)

My Projects

[Manage](#) [Completed](#) [Create New](#)

CRU-Programmers

CRU-Designers

CRU-Infrastructure

PM

Search:

Name		Contact	Unit	Complexity	Start	End	Deadline	Status		
		Agribusiness	Carrie Cloud	CAT	5	2/21/2011*	5/31/2011*	5/31/2011	Developing	
		Nutrition Goals Phase 1	Larissa Leavens	Nutrition Department	4	2/21/2011*	7/31/2011*	3/31/2011	Beta	
		Grants Management	DeeDee Kitterman	Executive	2	12/1/2010*	5/31/2011*	1/31/2011	Beta	
		MAAPS v2	Lili Byne	UAP	5	8/3/2009*	11/3/2009*	12/31/2020	Developing	
		SIMS-PETS v2	Scott Kirkland	UAP		n/a	n/a	n/a	Beta	
		Online Pre-Purchasing System	Pat Conners	CRU	8	7/11/2011*	n/a	2/29/2012	Developing	
		Generalized WebApp theme	Scott Kirkland	CRU	3	n/a	n/a	n/a	Beta	
		ESRAv2	Scott Kirkland	CRU	4	6/30/2011*	12/31/2011*	12/31/2011	Developing	
		Action Tracking	Fritz	CPT	1	n/a	n/a	n/a	Waiting	
		Data Dictionary	Ken Taylor	CRU	2	n/a	n/a	n/a	Waiting	
		Russell Ranch Datasets	Emma Torbert	ASI	4	n/a	n/a	n/a	Waiting	



Online Pre-Purchasing System

[Glossary](#) [Sections](#) [Requirements](#) [Use Cases](#) [Files](#) [Tasks](#) [Back to List](#)

Project

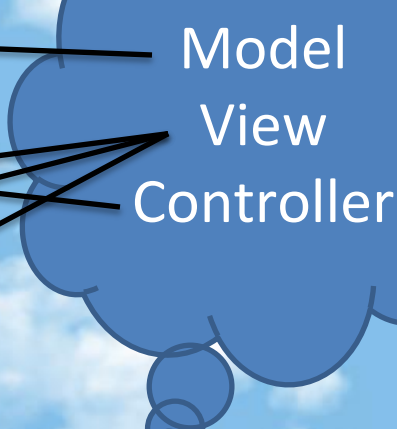
Complexity	8
Deadline	2/29/2012
Begin	7/11/2011*
End	n/a
Contact	Pat Conners
ContactEmail	pmconners@ucdavis.edu
Unit	CRU
Date Added	2/23/2010 9:09 AM
Last Update	8/3/2011 3:53 PM
Workgroup(s)	CRU-Programmers
Is Public	Yes

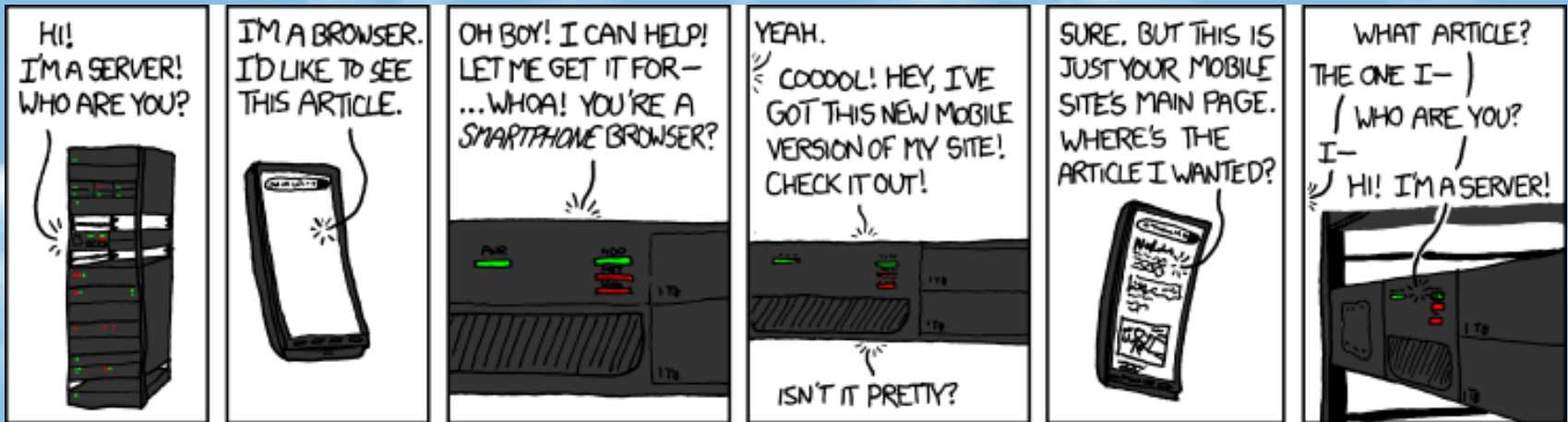
[Generate SRS](#)[Generate Draft SRS](#)

Secure Agile Cloud Development

- We've been talking Agile, and Cloud, but this ~~is~~ isn't the IT ***Security*** Symposium after all
- Security comes from correct code
- Correct code comes from validation
- Validation comes from testing and measurement

Interlude: REST and MVC

- Client-server
 - Layered
 - Stateless
 - Cacheable
 - Uniform Interface
- 



Interlude: Best Practices

UCDArch Common Library (post Microsoft EL):

- MVC/REST
- JQueryUI(JQueryMobile) for rich UI
- Well-defined hooks for unit tests
- Client and Server side validation
 - One definition for both! (<http://dataannotationextensions.org/>)
 - @Html.ValidationSummary(true)
- Security
 - CAS (Shibboleth coming)
 - Roles (CatBert, roll your own)
 - Auto-protect against CSS!
 - @Html.AntiForgeryToken()
 - DataValidation 1x (renders to Javascript or back-end)
- Logging (ELMAH, Yammer Metrics)
- Data Access
 - Auto-mapping based on conventions
 - Fluent Nhibernate
 - @this.MultiSelect("User.Organizations").Options(Model.Organizations, x=>x.Id,x=>x.Name).Selected(Model.User.Organizations.Select(x=>x.Id))
- Available with NuGet! (<http://nuget.org/>) at <http://nugetgallery.caes.ucdavis.edu/>
- On GitHub (of course) <https://github.com/ucdavis/UCDArch>



<http://xkcd.com/934/>



acgetchell | Dashboard | Inbox 0 | Account Settings | Log Out

Explore GitHub | Gist | Blog | Help | Search...

ucdavis / UCDArch

Admin | Unwatch | Fork | Your Fork | Pull Request | 8 | 5

Source | Commits | Network | Pull Requests (0) | Fork Queue | Issues (2) | Wiki (4) | Graphs | Branch: master

Switch Branches (3) | Switch Tags (3) | Branch List

UC Davis Core Architecture using ASP.NET MVC, NHibernate, and Castle DI — Read more
<http://ucdarch.codeplex.com/>

Downloads

SSH | HTTP | Git Read-Only | `git@github.com:ucdavis/UCDArch.git` | Read+Write access

updated the mini profiler and completely integrated it with nhibernate s...



srkirkland (author)
July 29, 2011

commit 7417e61e3efe265ee789
tree 1fed9ed874d6314af672
parent deaec280b325a0f2080a

UCDArch /

name	age	message	history
SampleUCDArchApp/	July 29, 2011	updated the mini profiler and completely integrate... [srkirkland]	
UCDArch/	July 29, 2011	updated the mini profiler and completely integrate... [srkirkland]	
.gitignore	June 10, 2011	Added mini profiler and integrated into sample app [srkirkland]	
LICENSE.txt	February 08, 2011	Cleaned up formatting [acgetchell]	
README.txt	January 26, 2011	updated readme with migration path [srkirkland]	

README.txt

Project Description
UC Davis Core Architecture using ASP.NET MVC, NHibernate, and Castle DI.

The goal of this projects is to help developers produce excellent ASP.NET MVC applications using a strong framework proven under real-world conditions at UC Davis. UCDArch has a few specific portions which might be useful to UCDavis alone (like CAS Authentication), but the majority of the project could help others build strong MVC applications.

Note: This repository was migrated from <http://ucdarch.codeplex.com/> and only retains commit history since version 2.0. For previous commit history, see <http://ucdarch.codeplex.com/>



beta

Sign In

Home

Packages

Contribute

UC Davis: Jump Start Your Projects with NuGet

NuGet is a Visual Studio extension that makes it easy to install and update open source libraries and tools in Visual Studio.

When you use NuGet to install a package, it copies the library files to your solution and automatically updates your project (add references, change config files, etc). If you remove a package, NuGet reverses whatever changes it made so that no clutter is left.

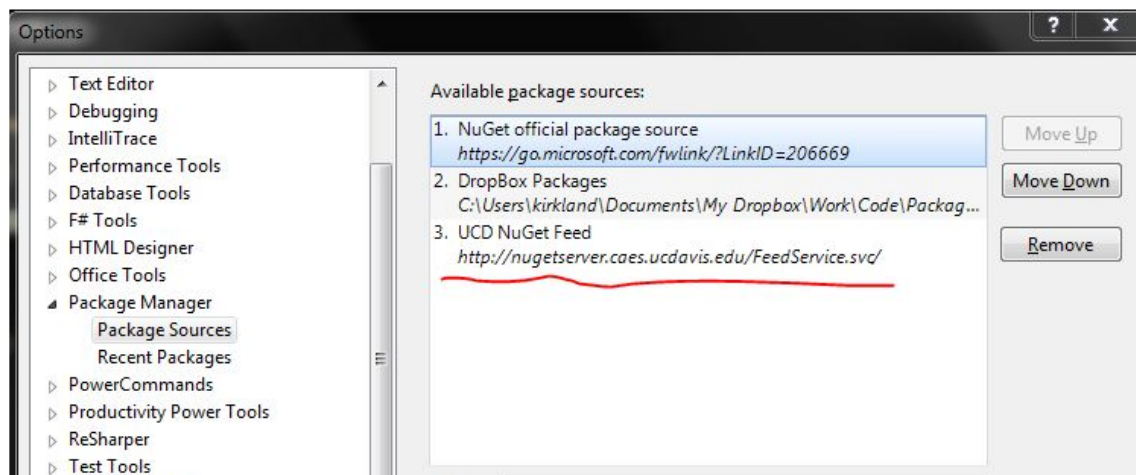
So *install NuGet* if you don't already have it, *search for packages*, and *get a jump on your next project!*

Important Notice: You can develop your own package and share it via the NuGet Gallery. For now, email Scott Kirkland (srkirkland at ucd...) if you would like an account in order to upload NuGet packages of your own for the benefit of UC Davis Application Developers.

Using the UC Davis NuGet Feed from Visual Studio 2010

The UCD NuGet feed is located at <http://nugetserver.caes.ucdavis.edu/FeedService.svc/>.

In Visual Studio, go to Tools--> Library Package Manager --> Package Manager Settings



Test-Driven Development

Big topic!

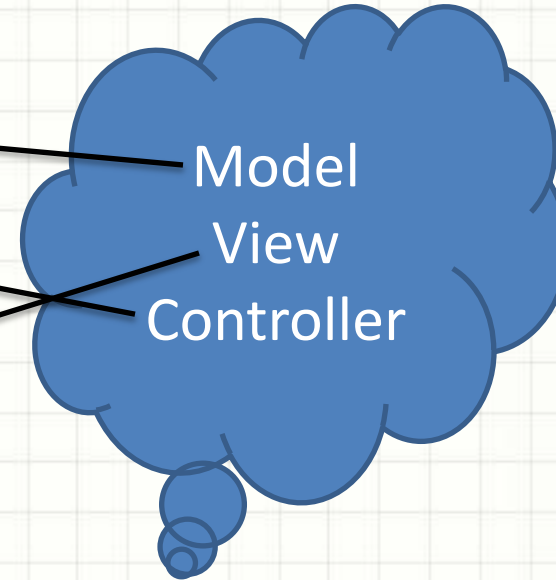
Basic idea: write functional tests that verify that things work as expected!

For a very thorough presentation, see Jason Sylvestre's "How To Write Unit Tests":

<https://github.com/jSylvestre/ResharperTemplates/blob/master/How%20To%20Write%20Unit%20Tests.pptx>

Types Of Unit Tests

- Repository Tests
- Controller Tests
- Interface Tests



Jason Sylvestre, How to Write Unit Tests,
<https://github.com/jSylvestre/ResharperTemplates>

Repository Tests

- Test Database actions
- Done on an in memory SQL Lite database created from the mapping file.
- This is done by using UCDArch's base class in testing "FluentRepositoryTestBase"

Controller Tests

- Mapping Tests
- Boundary Tests of all public methods
- Reflection Tests

Jason Sylvestre, How to Write Unit Tests,
<https://github.com/jSylvestre/ResharperTemplates>

Interface Tests

- Sometimes these need to be run within the structure of a Controller test
- Check that expected parameters are passed
- Check that any actions or return values are what is expected
- Even if you don't write tests for your interfaces, you just have one place to review the logic.

Jason Sylvestre, How to Write Unit Tests,
<https://github.com/jSylvestre/ResharperTemplates>

Stages of Agile Cloud Development

1. Developing using cloud-based source control management (SCM) repositories
2. Build in the cloud using virtual images, provide feedback to continuous integration (CI) servers
3. Testing in the cloud by building test machines
4. Production in the cloud, automated deployment, load-balancing, analytics

Projects -- TeamCity

builder/

acgetchell's Zotero GitHub Dropbox Mint.com Toodledo Delicious StrongLifts GymnasticBodies

Welcome, Guest user Logout

Projects Agents (1) Build Queue (0)

Projects

Collapse All | Expand All 0 build(s) running.

Agribusiness

Debug

#1.99 Tests passed: 19 No artifacts alan lai <anlai... (3) 07 Jun 11 16:03 (50s)

Catbert (Catbert Role Management System)

v3 -- Release Pending (167)

#3.16 Tests passed: 20 No artifacts No changes 30 Nov 10 11:38 (1m:34s)

v4 -- Release

#4.185 Tests passed: 500, ignored: 1 No artifacts scott kirkland ... (1) 29 Apr 11 10:55 (2m:49s)

Commencement (Student Commencement Registration)

Debug

#1.732 Tests passed: 1270, ignored: 4 No artifacts jason sylvestre... (1) 17 May 11 10:54 (10m:40s)

CRP (Converence Registration and Payments)

Debug

#1.724 Tests failed: 5, passed: 2219 No artifacts aesdean\sylvest... (1) 19 May 11 07:43 (9m:10s)

Dogbert (Dogbert MVC)

Debug

#1.146 Success No artifacts aesdean\lai (1) 28 Feb 11 14:41 (24s)

EligibilityList

Debug

#2.114 Tests passed: 4 No artifacts scott kirkland ... (1) 09 Jun 11 11:25 (42s)

ESRA (Employee Salary Review Analysis)

Debug

Continuous Integration with Team City

<http://www.jetbrains.com/teamcity/>

“Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly.”

– Martin Fowler

TeamCity

- Free for small/medium teams (20 build configs) or open source projects
- Works with .NET, Java, and Ruby
- Integrates with GitHub, Perforce, Subversion, Team Foundation Server, etc.
- Easy to configure for CI (e.g. every time code is checked into a repo)
- History builds → quickly identify unsafe modifications, restore lost artifacts (files, etc.)
- Build infrastructure scales using Amazon's EC2 (configure virtual build agents to start instances using AMIs)

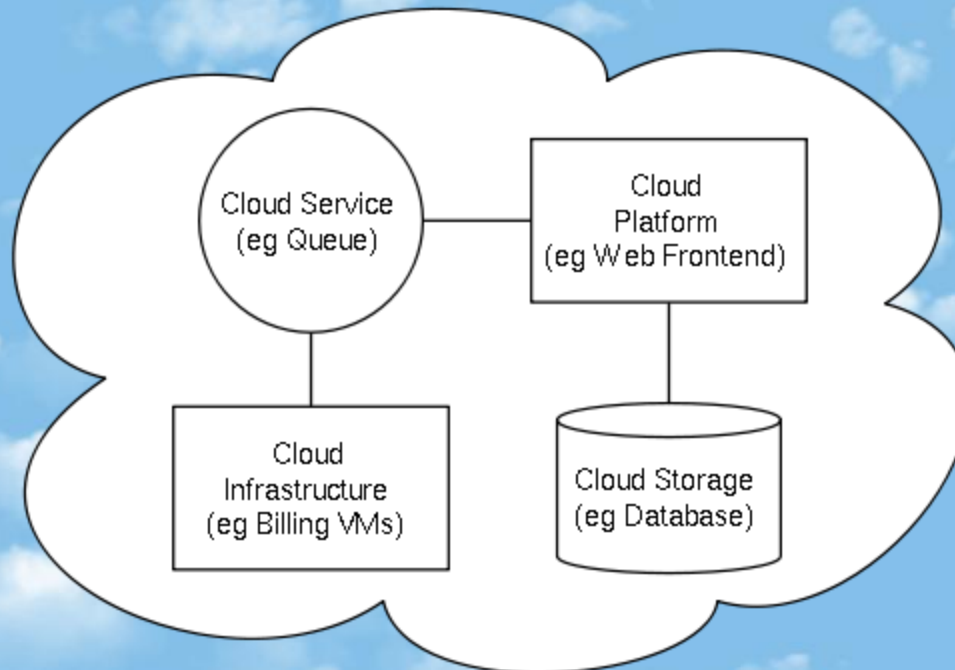
What have we accomplished?

- Simplicity
- Self-organizing teams
- Regular adaption to changing circumstances
- Projects are built around motivated, trusted individuals
- Sustainable development, able to maintain a constant pace
- Welcome changing requirements, even late in development
- Continuous attention to technical excellence and good design

Stages of Agile Cloud Development

1. Developing using cloud-based source control management (SCM) repositories
2. Build in the cloud using virtual images, provide feedback to continuous integration (CI) servers
3. Testing in the cloud by building test machines
4. Production in the cloud, automated deployment, load-balancing, analytics

Cloud Applications



Amazon EC2

<http://aws.amazon.com/about-aws/>

- High-profile failure
- Known-good redundancy methods, better monitoring now

The screenshot displays the AWS Management Console interface for the Amazon EC2 service. The browser address bar shows the URL <https://console.aws.amazon.com/ec2/home?region=us-east-1>. The console header includes the AWS logo, navigation links (AWS, Products, Developers, Community, Support, Account), and a welcome message for Adam Getchell. The main content area is titled "Amazon EC2 Console Dashboard" and is divided into several sections:

- Navigation:** A sidebar on the left with a "Region:" dropdown set to "US East (Virginia)". It lists various EC2-related links: EC2 Dashboard, INSTANCES (Instances, Spot Requests, Reserved Instances), IMAGES (AMIs, Bundle Tasks), ELASTIC BLOCK STORE (Volumes, Snapshots), and NETWORKING & SECURITY (Security Groups, Elastic IPs, Placement Groups, Load Balancers, Key Pairs).
- Getting Started:** A central yellow box with the text "To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance." and a prominent "Launch Instance" button. A note below states: "Note: Your instances will launch in the US East (Virginia) region."
- My Resources:** A section on the right showing the current state of resources in the US East (Virginia) region. It includes a "Refresh" button and counts for: 0 Running Instances, 0 Elastic IPs, 0 EBS Volumes, 0 EBS Snapshots, 0 Key Pairs, 1 Security Group, 0 Load Balancers, and 0 Placement Groups.
- Service Health:** A section at the bottom left showing the "Current Status" of the Amazon EC2 service. It indicates that the service is operating normally in the US East - N. Virginia region, with a link to "View complete service health details".
- Related Links:** A section on the bottom right providing links to Documentation, All EC2 Resources, Forums, Feedback, and Report an Issue.

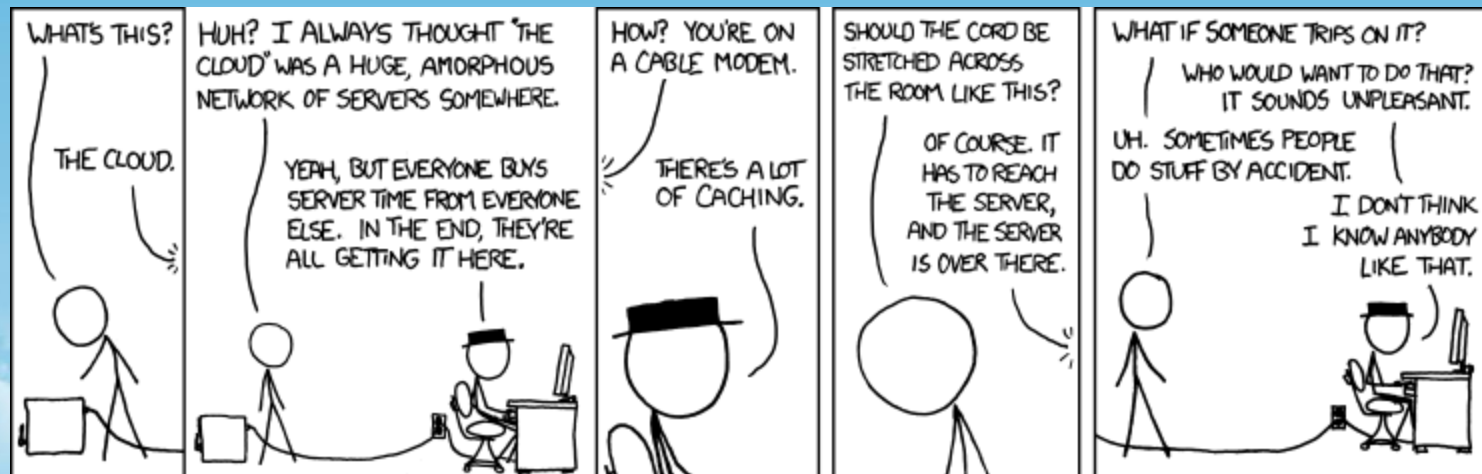
The footer of the console contains copyright information (© 2008 - 2011, Amazon Web Services LLC or its affiliates. All right reserved.) and links to Feedback, Support, Privacy Policy, and Terms of Use, along with the Amazon logo and the text "An amazon.com. company".

We tried Amazon first

Here's what we'll look for (from another vendor)

- Console access
- Regular and reliable backups
- Ability to quickly access and apply snapshots
- Disk imaging to allow rapid implementation (found AMIs difficult to work with in P2V)
- Fine grained security and firewall control
- smtp.ucdavis.edu authentication
- Good (fanatical?) customer support
- Community support

The Real Reason Amazon EC2/S3 Went Down



<http://xkcd.com/908/>

Azure

<http://www.microsoft.com/windowsazure/>

- .NET, Java (Hadoop), PHP and Ruby
- Used by Apple's iCloud*

The screenshot displays the Windows Azure Platform management console. The top navigation bar includes links for Billing, Adam Getchell, and Sign Out. The main interface is divided into several sections:

- Left Navigation Panel:** Contains links for Deployment Health, Affinity Groups, Management Certificates, Hosted Services (2), Storage Accounts (1), User Management, Home, Hosted Services, Storage Accounts & CDN, Database, Reporting, Service Bus, Access Control & Caching, and Virtual Network.
- Top Action Bar:** Includes buttons for New Hosted Service, New Production Deployment, New Staging Deployment, Upgrade, Configure, Delete, Start, Stop, Swap VIP, Configure OS, Reboot, Reimage, Enable, Configure, and Connect.
- Hosted Services Table:** A table listing hosted services with columns for Name, Type, Status, and Environment. The table shows a hierarchy starting with CAES-UCDAVIS (Subscription, Active), which includes AzureTestWithCE (Hosted Service, Created), Certificates, AzureSimple (Hosted Service, Created), Certificates, AzureTestSimp (Deployment, Ready, Staging), and MvcApplicati (Role, Ready, Staging). The MvcApplicati role has an instance named MvcApplic (Instance, Ready, Staging).
- Properties Panel:** Displays details for the selected service, including Account administrator (acgetchell@ucdavis.edu), Created date (1/4/2011 9:47:07 PM UTC), Name (CAES-UCDAVIS), Quota (Cores: 1 / 20, Hosted Services: 2 / 6, Storage: 1 / 5), Status (Active), Service administrator (acgetchell@ucdavis.edu), and Subscription ID.

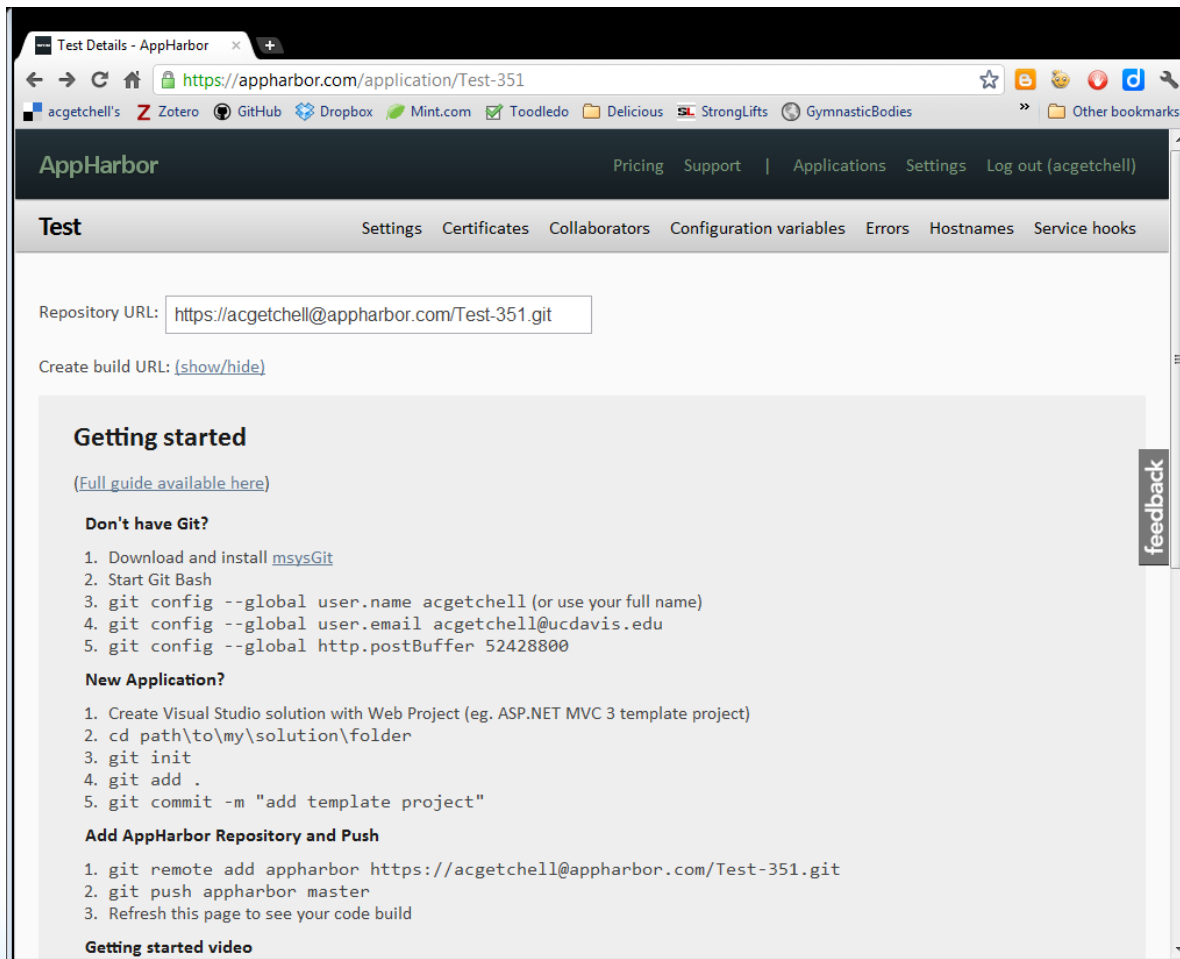
At the bottom, there is a status bar with a refresh message, a link to the old portal, copyright information (© 2010 Microsoft Corporation), and links to Privacy Statement, Terms of Use, Help and Support, and Feedback.

* <http://www.zdnet.com/blog/microsoft/is-apple-really-using-windows-azure-to-power-icloud/9687>

AppHarbor

<https://appharbor.com/>

- “Azure done right”
- .NET deployment integrated with Git



The screenshot shows a web browser window with the AppHarbor interface. The page title is "Test Details - AppHarbor". The address bar shows the URL <https://appharbor.com/application/Test-351>. The browser's bookmark bar includes links to acgetchell's, Zotero, GitHub, Dropbox, Mint.com, Toodledo, Delicious, StrongLifts, and GymnasticBodies. The AppHarbor header includes the logo and navigation links: Pricing, Support, Applications, Settings, and Log out (acgetchell). Below the header, a sub-header contains "Test" and several tabs: Settings, Certificates, Collaborators, Configuration variables, Errors, Hostnames, and Service hooks. The main content area displays the "Repository URL" as `https://acgetchell@appharbor.com/Test-351.git` and a "Create build URL" link with a "(show/hide)" toggle. A "Getting started" section follows, with a link to the "Full guide available here". It includes two sub-sections: "Don't have Git?" with a list of steps to install and configure Git, and "New Application?" with steps to create a Visual Studio solution and push it to AppHarbor. A "Getting started video" link is at the bottom. A vertical "feedback" button is located on the right side of the page.

Test Details - AppHarbor

<https://appharbor.com/application/Test-351>

acgetchell's Zotero GitHub Dropbox Mint.com Toodledo Delicious StrongLifts GymnasticBodies Other bookmarks

AppHarbor Pricing Support Applications Settings Log out (acgetchell)

Test Settings Certificates Collaborators Configuration variables Errors Hostnames Service hooks

Repository URL: `https://acgetchell@appharbor.com/Test-351.git`

Create build URL: [\(show/hide\)](#)

Getting started

[\(Full guide available here\)](#)

Don't have Git?

1. Download and install [msysGit](#)
2. Start Git Bash
3. `git config --global user.name acgetchell` (or use your full name)
4. `git config --global user.email acgetchell@ucdavis.edu`
5. `git config --global http.postBuffer 52428800`

New Application?

1. Create Visual Studio solution with Web Project (eg. ASP.NET MVC 3 template project)
2. `cd path\to\my\solution\folder`
3. `git init`
4. `git add .`
5. `git commit -m "add template project"`

Add AppHarbor Repository and Push

1. `git remote add appharbor https://acgetchell@appharbor.com/Test-351.git`
2. `git push appharbor master`
3. Refresh this page to see your code build

Getting started video

feedback

Other Cloud Application hosting

- Rackspace (<http://www.rackspace.com/>)
 - Cloud Servers (virtual instance)
 - \$0.08/hour/1GB RAM & 40GB disk
 - OpenStack for private clouds
 - <http://www.rackspace.com/cloudbuilders/openstack/>
- Heroku (<http://www.heroku.com>)
 - Ruby, Node.js
 - Git friendly
 - \$0.05/hour/web process (\$36/month)
 - \$.05/hour/background process
- Engine Yard (<http://www.engineyard.com/>)
 - Ruby
 - GitHub friendly
 - Production Application (2 medium load-balance app instances, 1 medium database instance) ~\$534.00/month

Nice, but what about the data?

Can we store public data in the public cloud?

Can we store business data in the public cloud?

Can we store Restricted Information* in the public cloud?

“Restricted information describes any confidential or personal information that is protected by law or policy and that requires the highest level of access control and security protection, whether in storage or in transit. The term “restricted” should not be confused with that used by the UC managed national laboratories where federal programs may employ a different classification scheme.”

* <http://www.ucop.edu/ucophome/policies/bfb/is3.pdf>

Nice, but what about the data?

~~Can~~ May we store public data in the public cloud?

~~Can~~ May we store business data in the public cloud?

~~Can~~ May we store Restricted Information in the public cloud?

Policy Exception so far (March-September 2011):

- No Restricted Information shall be remotely available in Azure (cloud services)

Cloud Data hosting

Relational Database Management
Systems RDBMS vs. Key-Value
("NoSQL")

Why do Clouds prefer NoSQL?

RDBMS

Provides:

- Atomicity: All operations in a transaction will complete, or none will
- Consistency: The database will be in a consistent state when the transaction begins and ends
- Isolation: The transaction will behave as if it is the only operation being performed upon the database
- Durability: Upon completion of the transaction, the operation will not be reversed

Brewer's CAP Theorem*

Web services cannot ensure all three of the following properties at once:

- Consistency: The client perceives that a set of operations has occurred all at once
- Availability: Every operation must terminate in an intended response
- Partition tolerance: Operations will complete even if individual components are unavailable

In order to scale, we must have partitioning and availability. Thus we must give up consistency.

* <http://citeseer.ist.psu.edu/544596.html>

BASE vs. ACID*

BASE is Basically Available, Soft State, Eventually Consistent

- Essentially, you must find ways to relax your constraints
- For example, if you have a ticket selling application, you must assume that there may be tickets that have been bought but not yet recorded as purchased
- Application errors = data errors
 - RDBMS are internally consistent, by contrast

* <http://queue.acm.org/detail.cfm?id=1394128>

Why NoSQL?*

- BASE allows scalability for clouds
- No Entity Joins (data is denormalized), so faster
- Data model matches data structures in code better

* <http://www.readwriteweb.com/enterprise/2009/02/is-the-relational-database-doomed.php>



Think carefully about whether you truly need
Relational Data

It will be more expensive and slower

But, so far we have not needed to scale out
drastically and have stuck with RDBMS

Cloud NoSQL data sources

- Amazon SimpleDB (<http://aws.amazon.com/simplifiedb/>)
 - Automatic geo-redundant replication within Region
 - Automatic indexing
 - Simple API
- Redis (<http://redis.io/>)
 - Open-source (powers GitHub)
(<http://code.google.com/p/redis/>)
 - Excellent API for programmer
 - Dozens of clients for most programming languages
- (Didn't mention Google AppEngine Data Store, because you can't use it outside of AppEngine.)

Cloud SQL data sources

- Microsoft SQL Azure
(<http://www.microsoft.com/windowsazure/sqlazure/database/>)
 - High-availability and fault tolerance
 - T-SQL syntax
- AppHarbor MySQL or Microsoft SQL Server
 - Shared servers \$10-10GB/month, dedicated server costs extra
- Amazon EC2 Running Microsoft Windows Server & SQL Server
 - No change for SQL/.NET developers
- Rackspace Cloud Server running Microsoft SQL Server 2008 R2

Agile Development Feedback

- Working software is the principal measure of progress
- Customer satisfaction by rapid delivery of useful software
- Close co-operation between business people and developers
- Face-to-face conversations as the best form of communication
- Working software is delivered frequently (weeks rather than months)

There was a Dog..B-I-N-G-O

Waiting

Winner

Game 10

8

7

2

We love your feedback!

Your feedback is important to us

Feedback [browse ideas→](#)


-  **Excellent program**
4 votes · 1 comment
-  **Beep when a new ball drops**
3 votes · 0 comments
-  **The game admin page does not show ...**
8 votes · 0 comments

Share an idea

Share your idea in our Bingo forum so others can vote for it. If we find a similar idea to yours, you can vote for it instead.

How can we improve Bingo?

Signed in as adam <automatedemail@caes.ucdavis.edu>

powered by  uservoice

UserVoice

- Single Sign-on (we integrated it with CAS)
- Public API
- iPhone SDK
- Facebook Pages
- Domain aliasing
- Custom logo & colors
- CSS customization
- 10 forums
 - File attachment
 - Private forums
- Feedback Widget
- Excel export
- \$45/month

Measure Your Code

See Coda Hale's "Metrics, Metrics Everywhere"

- <http://codahale.com/codeconf-2011-04-09-metrics-metrics-everywhere.pdf>



<http://xkcd.com/371/>

Coda Hale's Rap

- Code is business value
- Business value is new features, improved features, fewer bugs, making future changes easier, adding unit tests
- Make better decisions about code
- Code generates business value when it *runs*, not when it is *written*

Coda Hale's Rap

- Thus, we need to know what code is doing when it is running
- We need to measure it when it is running and generating business value in Production
- We have a mental model of the code, but it is often wrong

Coda Hale's Rap

- Thus, we need to know what code is doing when it is running
- We need to measure it when it is running and generating business value in Production
- We have a mental model of the code, but it is often wrong
- We also have to measure the right things

Yammer Metrics for Java/Scala

- Gauges
 - Instantaneous readings
- Counters
 - 64-bit integer with increment/decrement
- Meters
 - Increment-only counters which track mean rates and exponentially-weighted moving average (same formula as UNIX `la`)
- Histograms
 - Distribution measurements: count, max, min, mean, standard deviation, median, 75/95/98/99/99.9 %
- Timers
 - Duration and rate information, plus histogram info

Yammer Metrics for Java/Scala

- <https://github.com/codahale/metrics>

.NET/CLR version

- <https://github.com/danielcrennan/metrics-net>

Local fork

- <https://github.com/ucdavis/metrics-net>

Ruby version

- <https://github.com/johnewart/ruby-metrics>

References

- http://en.wikipedia.org/wiki/Agile_development
- Image on Slide 4 was uploaded to Wikipedia on 2010-07-02 18:40 by [Dbenson](#) 2700×3600× (906118 bytes) *This poster provides a good visual of the standard Agile Software Development methodology. This original work was created by VersionOne, Inc. and is licensed under the Creative Commons Attribution CC-BY-SA license*
- http://en.wikipedia.org/wiki/Cloud_computing
- Image on Slide 8 was uploaded to Wikipedia and the public domain on 2011-02-15 by Benjamin P. Griner and Philip J. Butler.
- <http://git-scm.com/about>
- “Hitler and Cloud Computing Security”
<http://www.youtube.com/watch?v=VjfaCoA2sQk>

Agile, Secure Cloud Application Development & Management

Adam Getchell

acgetchell@ucdavis.edu

<https://github.com/acgetchell>

<http://www.linkedin.com/in/adamgetchell>

ucdotnet@ucdavis.edu

UC Computing Systems Conference

August 7-9, 2011

University of California Merced