Software Development Projects

Working in an agile environment (TDD, BDD, DDD Pair Programming etc.)

I hope this gives you a good outline on some of the major contributions I have been able to provide as part of my roles in each company. All images shown are publically available to anyone that has purchased these products and shown here for illustrative purposes only.

HRand/Medi-Direct Inc.

SENIOR SOFTWARE DEVELOPER/ARCHITECT

As part of architecting and developing the system I have been a ferocious learner and tried out many different technologies and techniques before coming up with a framework that is truly pleasing to the eye and is extremely easy to work with.

I work hard to ensure developers have excellent tools and helpers to ensure they can develop in a concise and consistent manner by hiding complexities under intuitive interfaces that are well documented and fully tested.

Because of this undertaken we as a team find it very easy to refactor major pieces of our codebase without having to worry about breaking anything due to how much API and maybe more importantly UI testing that hits every piece of functionality and makes things really scalable for bringing on new developers and will allows us to grow fast.

BEHAVIOUR DRIVEN DEVELOPMENT AUTOMATED TESTING FRAMEWORK

Background:

Early on in the project it was decided that we would develop with the Behavior Driven Development process. The idea was to have tests to ensure we are building the product in a way that is transparent and understandable to our stakeholders.

Motivation:

I was finding that there was a massive amount of boilerplate and lack of maintenance when we initially started creating API and UI tests against static Behaviors. Early on I recognized this as a risk and collaborated with the team and it was agreed that I would come up with an automation process to speed up development overall and address the risk of having separate processes for the API and UI.

Sample Feature File

I have been able to take the idea of Behavior Driven Development to the next level of automation by building a framework that does some of the following:

- All steps are implemented in a generic way in order to promote re-use across any number of resources.
- Converts background steps into fixture data which can be referenced in any other background/scenario step for convenience.
- Detailed instruction on the syntax for each step so that developers/business can come up with clean scenarios.
- A request is built up based on given input from each step which is then sent to the API endpoint and there is an automated process to verify that the data sent was actually saved correctly and audited.
- The same request is used based on changes made and is then used to control the UI and again there is a check to ensure everything is saved and audited as expected.

Sample Step Implementation

Some outcomes:

- Provides developers with a clear and concise interface that hides the complexities so they can focus a lot more the task at hand
- Extremely pluggable and adaptive to handling new resources and UI pages with minimal effort
- Provided helper methods that interact with the UI automatically based on the step e.g. if the step includes a reference it knows it will be updating an existing field.
- Simply fun to use and unbelievably well received.

Hubstaff Inc.

SENIOR SOFTWARE ARCHITECT

My responsibilities as a software architect was to take high level requirements and come up with functional & technical requirements for each team of developers to undertake. I was working directly with the CTO and Lead architect and was able to get through an intensive trial.

Both projects (timesheets & time off features) were successfully completed within the quarter and at the same time I was working on my own side projects.

TIME OFF

The high level requirements here were to implement a system to allow for the creation of public holidays (paid & unpaid) and time off policies including unpaid, hourly & yearly accruals. In order to come up with a comprehensive document I did a lot of research on the problem at hand and also when through many competitors that with successful products and after multiple iterations landed on a final design. This included mockups & workflow diagrams.

TIMESHEETS

The high level requirements here were to create timesheets based off of activity recorded via Hubstaff client applications. Timesheets would display details of the activity including any prolonged idle time with each member with timesheets enabled expected to submit each based on pay schedule. What made this more complex was the interactions with other systems including invoicing, billing and scheduling. With a lot of effort, collaboration and iteration I was able to come up with a detailed and approved design.

SINGLE SIGN ON (FRONT AND BACK CHANNEL LOGOUT)

One of the first projects I worked on as part of my trial period was to close up potential security holes with the pre-existing SSO system. This included implementing front channel logout based on the official Open ID specification. This was used when a normal logout is performed to ensure that all sessions under the same browser was invalidated.

The second piece was to implement the back channel logout to invalidate all sessions regardless of browser by issuing a request for the identity provider to request each relying party to sign out completely.

Cisco (July 2014 – October 2017)

SENIOR SOFTWARE / PRODUCT DEVELOPER

All of our major features were featured flagged to allow for continuous integration and quick rollback if a significant bug is found on production especially due to the sensitive nature of the features we were developing.

MODULARIZE AUTHORIZATION

One of the senior developers on extracting our login process into a separate engine and now it serves as a micro service for use across multiple applications. As part of this work we also introduced an SSO option and our own Identity Provider.

INTRODUCED PARTNER ACCOUNTS

Lead developer & team lead on a brand-new feature that now allows partner companies to service their clients and produce accurate reporting like never before. This is one of the highest priority projects of the year and is expected to bring in a lot of revenue

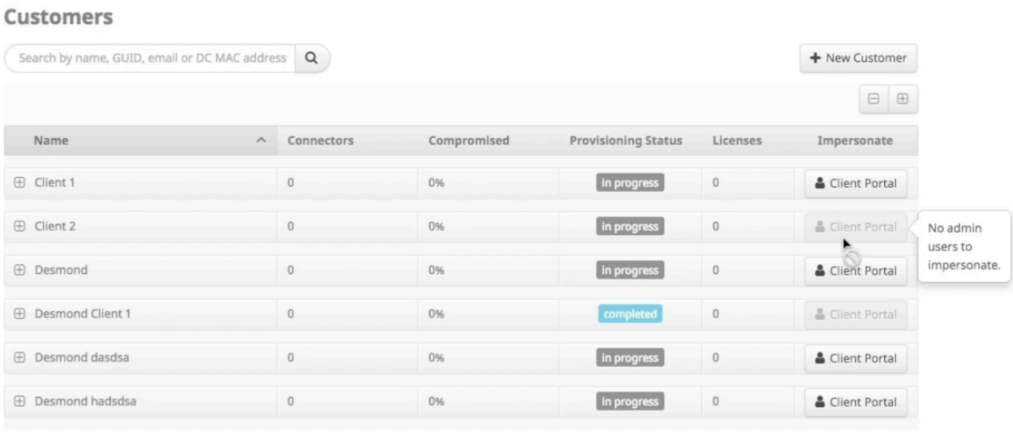
- Introduced Single Table Inheritance (STI) to differentiate major account types. Provided good encapsulation and separation of concerns.
- Devised a daily report that records summary information on all accounts sourced from MySQL, Cassandra and MongoDB

- Used the daily report to produce aggregate reports on individual partners including breakdown per customer

- Came up with a Partner Dashboard that gives them a quick look into the health of their customers

- In order to manage individual accounts, we offered the impersonation feature to allow them to manage customer accounts on behalf of the customer

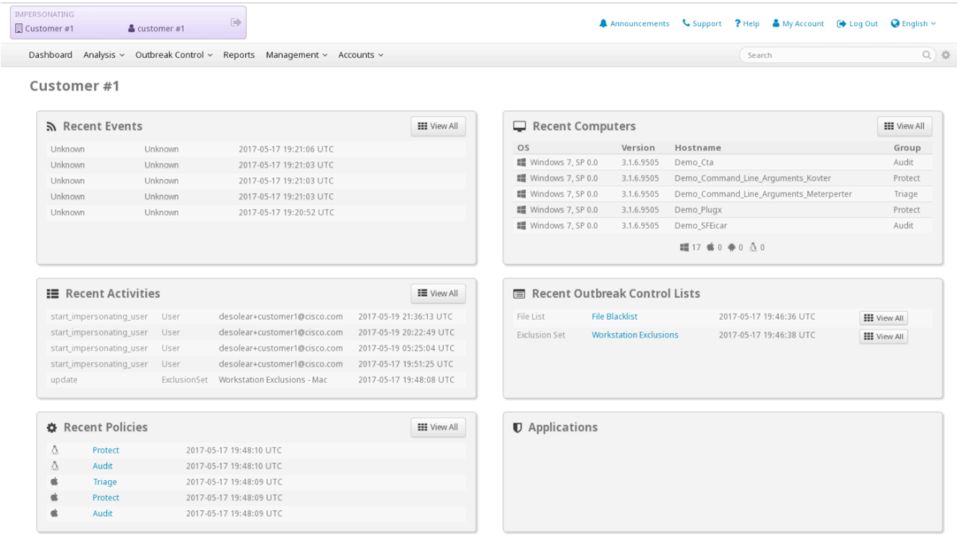
Partner Customer Dashboard



Name	Connectors	Compromised	Provisioning Status	Licenses	Impersonate
⊞ Client 1	0	0%	in progress	0	Client Portal
⊞ Client 2	0	0%	in progress	0	Client Portal
⊞ Desmond	0	0%	in progress	0	Client Portal
⊞ Desmond Client 1	0	0%	completed	0	Client Portal
⊞ Desmond dasdsa	0	0%	in progress	0	Client Portal
⊞ Desmond hadsdsa	0	0%	in progress	0	Client Portal

No admin users to impersonate.

Individual Partner Customer Overview



Customer #1

Recent Events

Unknown	Unknown	2017-05-17 19:21:06 UTC
Unknown	Unknown	2017-05-17 19:21:03 UTC
Unknown	Unknown	2017-05-17 19:21:03 UTC
Unknown	Unknown	2017-05-17 19:21:03 UTC
Unknown	Unknown	2017-05-17 19:20:52 UTC

Recent Computers

OS	Version	Hostname	Group
Windows 7, SP 0.0	3.1.6.9505	Demo_Cta	Audit
Windows 7, SP 0.0	3.1.6.9505	Demo_Command_Line_Arguments_Kovter	Protect
Windows 7, SP 0.0	3.1.6.9505	Demo_Command_Line_Arguments_Meterperter	Triage
Windows 7, SP 0.0	3.1.6.9505	Demo_Plugx	Protect
Windows 7, SP 0.0	3.1.6.9505	Demo_SFEicar	Audit

17 0 0 0

Recent Activities

start_impersonating_user	User	desolear+customer1@cisco.com	2017-05-19 21:36:13 UTC
start_impersonating_user	User	desolear+customer1@cisco.com	2017-05-19 20:22:49 UTC
start_impersonating_user	User	desolear+customer1@cisco.com	2017-05-19 05:25:04 UTC
start_impersonating_user	User	desolear+customer1@cisco.com	2017-05-17 19:51:25 UTC
update	ExclusionSet	Workstation Exclusions - Mac	2017-05-17 19:48:08 UTC

Recent Outbreak Control Lists

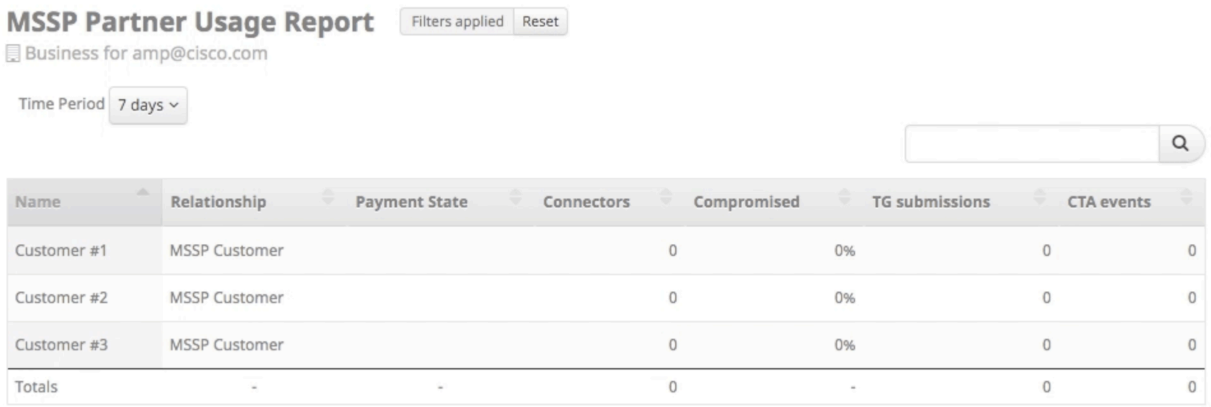
File List	File Blacklist	2017-05-17 19:46:36 UTC	View All
Exclusion Set	Workstation Exclusions	2017-05-17 19:46:38 UTC	View All

Recent Policies

Protect	2017-05-17 19:48:10 UTC
Audit	2017-05-17 19:48:10 UTC
Triage	2017-05-17 19:48:09 UTC
Protect	2017-05-17 19:48:09 UTC
Audit	2017-05-17 19:48:09 UTC

Applications

Usage Reports



MSSP Partner Usage Report

Business for amp@cisco.com

Time Period: 7 days

Name	Relationship	Payment State	Connectors	Compromised	TG submissions	CTA events
Customer #1	MSSP Customer		0	0%	0	0
Customer #2	MSSP Customer		0	0%	0	0
Customer #3	MSSP Customer		0	0%	0	0
Totals	-	-	0	-	0	0

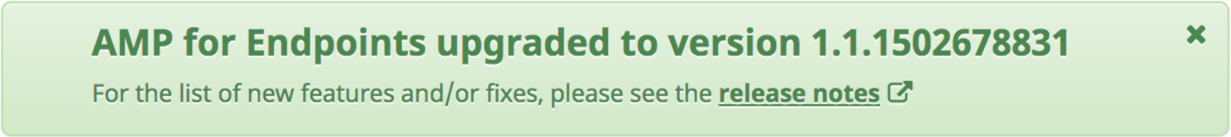
Global Announcements

Come up with Global Announcements

Backend Details

- Implemented on top of Starburst Gem Foundation
- Includes announcement views table in order to keep track of items read by user
- On click of /announcements all announcements are viewable and marked as read

Frontend Details



AMP for Endpoints upgraded to version 1.1.1502678831

For the list of new features and/or fixes, please see the release notes

- Announcements (General, Maintenance [**very rare**] and Upgrades as shown above)
- Includes preview as you type

Business Creation Revamp

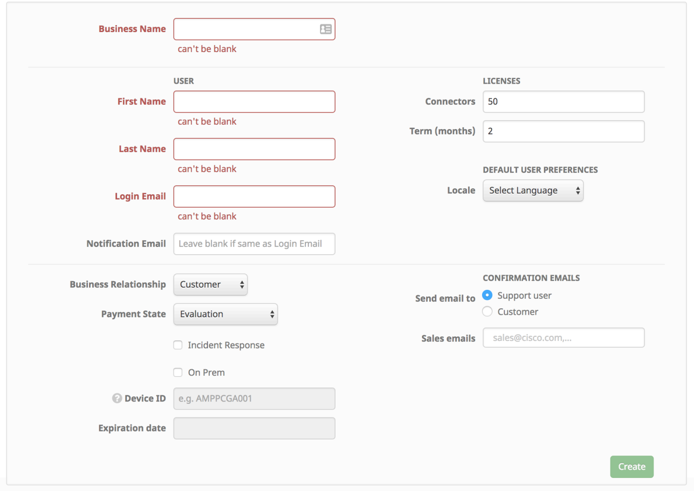
Project to revamp how we provision user accounts

Backend Details

Biggest issue here we've found is DB locks
- Figure out where the dead locks are coming from
- For us it was the provisioning code that was causing major issues
  - Went from old attributes tables to having one table with many columns
  - This was done in stages
  - Removed redundant items based on info on prod, many became default values that never change so no need for DB entries

- Done piece by piece until complete
- Moved provisioning to Delayed Job and ensured it worked much the same as before
- Improved the experience for the user and allowed them to do a lot more work instead of creating multiple tabs

Frontend Details



Business Name
can't be blank

USER

First Name
can't be blank

Last Name
can't be blank

Login Email
can't be blank

Notification Email: Leave blank if same as Login Email

LICENSES

Connectors: 50

Term (months): 2

DEFAULT USER PREFERENCES

Locale: Select Language

Business Relationship: Customer

Payment State: Evaluation

□ Incident Response

□ On Prem

Device ID: e.g. AMPPCGA001

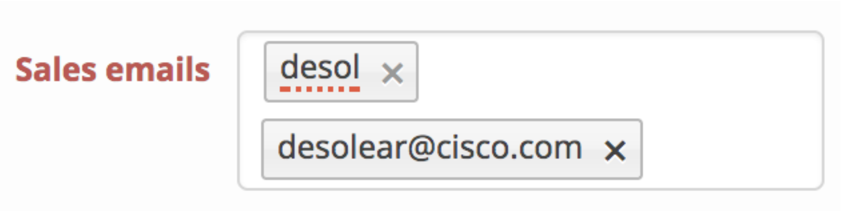
Expiration date

CONFIRMATION EMAILS

Send email to ● Support user ○ Customer

Sales emails: sales@cisco.com,...

- Unified error handling / client and server side validation looks the same whether on the form or alert



Sales emails

desol ×

desolear@cisco.com ×

Instead of just coming up with the standard fare I went a little further and still managed to deliver on time and customers loved it

- including email syntax highlighting
- validation is done using email validator that can handle multiple fields
- one click of item it goes into editing mode immediately

- x icon is highlighted on hover

- Inspired by Gmail’s version

Impersonation

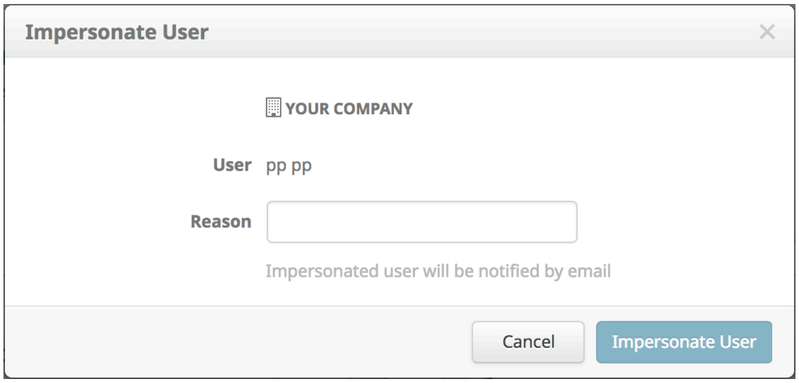
Allow our support staff to impersonate customer accounts

Backend Details

- Found out how this is done in the wild and tested many implementations until I landed with pretender gem and implemented changes on top of this
- Audit log all actions taken by the impersonating user
- All actions by the impersonating user displayed for the customer to see
- In order for a customer to opt into this feature they had to accept an agreement

Front end

- Support admin only
- User has to provide reason
- Came up with widget to make it obvious we are impersonating another user



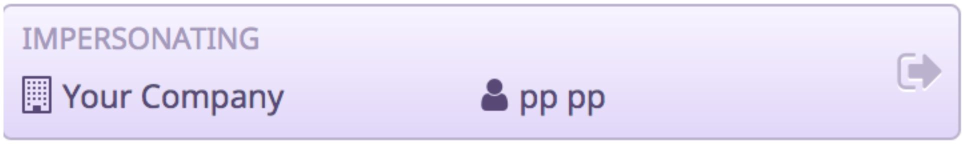
Impersonate User

YOUR COMPANY

User pp pp

Reason

Impersonated user will be notified by email



IMPERSONATING

Your Company

pp pp

Rails 4 & Ruby Upgrade

Lead developer on Rail 4 Upgrade. Had to be done very carefully to ensure there was no customer impact. Ruby upgrade was done with the use of feature flag and only globally enabled once all specs were passing for both Ruby versions. Rails 4 upgrade was done in a similar fashion. Ended up with no customer issues.

- Beefed up tests to ensure functionality works the same before and after the upgrade
- Replaced deprecated features
- Upgrade gems
- Fix broken specs

Bootstrap 3 & site wide localization

Lead developer on our new multi-language offering which involves localizing text across the application including handling of constants stored in the database.

Heavily involved in migrated from Bootstrap 2.x to 3.x without any production issues raised. Genuine massive undertaken that was achieved due to excellent collaboration between multiple teams.

This gave us the opportunity to clean-up the overall UI to use the newer look and feel that we introduced throughout the application. This was done with extensive collaboration with multiple team members and we came up with FAQ wiki and task list to complete the work on time and without any customer issues.

- Created a FAQ wiki in order to keep learnings / tips in the one place (this made it easier to bring others on to the team)
- Used resources that helped pinpoint necessary changes from bootstrap 2 to 3
- Feature flagged
- Formulated another FAQ for tips on adding text to localization files, this helped tremendously in keeping the workload down to a minimum for the translators

Smart Technologies (July 2013 – July 2014)

SENIOR SOFTWARE ENGINEER

SUBSCRIPTION APPLICATION (GOOGLE APP ENGINE)

Technologies:

- Java, Gson, Guava
- Google Directory API (Integrate with Google Groups)
- Guice dependency injection framework
- Jersey (JAX-RS API) web container
- Flexera integration (Entitlement software)
- Grails build framework (SCSS compiling from multiple sources)
- JMock, PowerMocks
- Thymeleaf templating language
- JQuery, KnockoutJS (data-binding), Datatables
- Bootstrap 3.x, 2.x
- HTML 5, CSS3 and SCSS
- Mercurial and Sub-Repositories (Developer style guide integration)

Context: Subscription app is a public user interface that allows Domain Administers to register for SMART amp and configure their domains in order to provision teachers and students to use the SMART amp product line.

1. Responsible for the overall architectural design and implementation
2. In preparation developed an amp wide “developer” style guide to complement the “UX” style guide to allow for UI components to be reused throughout all platforms to ensure alignment and allow for changes to be reflected in an efficient and consistent manner.
3. Fully developed all layers of the application from the back-end to the frontend liaising with team mates (OAuth integration, Flexera integration)

UI was developed using the latest technologies and looks stunning IMHO

Solium (July 2011 – July 2013)

SENIOR SOFTWARE ENGINEER

RECONCILIATION PLATFORM

Technologies:

- Ruby on Rails
- Rake
- JQuery, JQuery Tools, Datatables
- CSS
- PostgreSQL
- NewRelic performance management and monitoring software

1. Lead developer responsible for the overall architectural design and implementation
2. Upgraded application from Rails 2 to 3.x
3. Introduced C.I. and increased code-coverage from 20% to 85% which included refactoring controller heavy logic into appropriate models
4. With the use of NewRelic increased performance on most utilized screens with the use of efficient SQL and caching
5. Developed automated processes to reconcile both Share Purchase and Option/Award accounts daily (+1600) complimented with a user friendly U.I. which offered extended functionality and promoted reuse throughout

AUTO-IMPORT BROKER FILES

Technologies:

- Rails
- FTPFXP API (provides SSL/TLS connection capabilities)
- Jira API
- Crontab
- JQuery Tools (Dateinput)

1. Created a process that runs daily and logs into the designated SFTP site and performs a lookup to check if the latest file exists. If required file is found then the application proceeds to upload file and persist into the database in a normalized form.
2. Implementation includes an abstraction to the 3rd party FTP and Jira API's
3. Provided a user interface to manually import broker files into the system when the automated process does not succeed for any reason with the user being notified via Jira.
4. Implemented using the latest JQuery technologies and techniques

AUTO-RECONCILE CONTRIBUTION FILES

Technologies:

- Rails
- Soap4r API

1. Created an automated process that runs every 30 minutes and picks up all contribution files ready to be committed via abstracted Shareworks API.
2. For each contribution file there is a check to whether there exists matching transaction(s) then proceeds to commit the file into Shareworks.
3. If no match is found the assigned account administrator is notified via Jira to manually match via a user-friendly interface provided with the further ability to manually commit the files into Shareworks and into Wrecker simultaneously that includes full auditing of actions.

CRITICAL MASS (FEBRUARY 2011 – JULY 2011)

SOFTWARE DEVELOPER

WEBSITE DATA LAYER PROVIDER

Technologies:

- Grails web application platform
- Maven/Ivy
- JCR API

1. Lead developer responsible over the architectural design and implementation of four separate applications with many integration points.
2. With a focus on reuse, maintenance and clear design principles to allow for a uniform development process going forward
3. Showed excellent collaboration with all client externally/internally and produced a applications which meet and exceeded clients expectations.

SHAW COMMUNICATIONS (MAY 2008 – FEBRUARY 2011)

SOFTWARE DEVELOPER

RESTFUL INTEGRATION WEB SERVICE

Technologies:

1. Windows Communication Foundation (WCF)
2. Dependency injection, Inversion of Control (Castle Windsor Container)
3. Aspect Oriented Programming (Castle Core Interceptors) Cross-cutting concerns including logging and security
4. Castle Dynamic Proxy (Enabling interceptors to be associated to defined objects)

5. ADO.NET calls to Attunity server
6. Lead developer responsible over the architectural design and implementation of the Integration service.
7. This was an operational requirement to abstract integration with the legacy VAX-BASIC system from the rest of the SOA services so they can call this one service directly.
8. Collaborated with the legacy team and clients to come up with an appropriate platform and act as a pass-through layer.
9. Introduced Automated Continuous Integration and Automated deployment.

RESTful SOA Services

Technologies:

- Windows Communication Foundation (WCF)
- Dependency injection, Inversion of Control (Castle Windsor Container)
- Aspect Oriented Programming (Castle Core Interceptors) Cross-cutting concerns including logging and security
- Object Relational Mapping (ActiveRecord)
- NVelocity (Adding recursion support using custom directives)
- Hypermedia workflow ('ST' in REST)
- Url Rewriting (Helicon ISAPI Filter)

1. Working on knowledge already gained I worked with a great team to produce a very RESTful set of SOA services.

Interactive Enterprise, Ireland (September 2005 - May 2008)

Junior Software developer

CDR Payment script

Technologies:

- Python scripting
- JSP

1. Successfully created a script that processed CDR's from Oracle XML Data store and based on business logic aggregated payments and then handled calls to CitiBank payment gateway through a JSP hosted Servlet, including black-listing and e-mail notification