Steven Higgan

Personal Information

- Telephone: +64 21 936 439
- Email: stevenh@mad.geek.nz
- Github: https://github.com/shiggan
- Link to this document: https://shiggan.github.io/cv/

Competencies

Languages and Technologies

ASP.NET (MVC, WebAPI, Core), C#, SQL, Powershell, Dynamics CRM, Visual Studio Team Services, GIT, TFS, Automated Building (continuous integration), Automated Releasing (continuous delivery), Microsoft SQL Server, Devops, Agile, Azure DevOps (*azdo*) and Waterfall

Methodologies

Client engagements are wide ranging and varying; not all practices are relevant for all clients or systems. As functioning in a service delivery organisation, I have a lot of experience in applying just the right amount of agility in order to respond to situations as they arise.

A core aspect of my approach to implementing DevOps practices in a new client or system Determining where they are at; the fundamental concepts that should be applied as a minimum; once the fundamentals are established and as a part of a continuous improvement process move the client and system to a place that is appropriate for the client and their particular needs.

I have successfully implemented a number of dev-ops practices with a wide range of customer systems including.

- Automated database and website changes into release environments Automated delivery particularly to databases implies a number of other dependent processes, starting at branching and/or feature flagging through proving a set of changes is fit for purpose. Using Azure-Devops I have developed build, test, and release pipelines to deliver changes into systems. I enjoy demoing this process with clients new to the technology and ideas, there was a time when a client was visibly excited when I showed them the automation around approval workflows, particularly when also using the product backlog integration.
- Setting up continuous testing with reporting Where needed and appropriate separating unit and integration testing (e.g. integration testing on a deployed environment); also where needed setup code coverage reporting.

- Establishing branching strategy Sometimes in the role of maintaining systems you get your hands on a system where either there is no strategy in use; and the solution is not setup for continuous delivery and you are running into surprises in production, or the strategy that is being used is not working for any number of reasons. Back in the TFS days I worked on a system that employed a 'trunk development' and 'release environment' models in short you developed large features on trunk and branched the solution for each isolated-environment you went into (user-acceptance and production) and ported fixes back and forth. The features were large and complex; acceptance testing was slow and tough; with a bit of code churn and branch maintenance stress. We ended up moving to a 'release branch' and 'feature branch' model. We worked on features on separate branches (we could have used feature-flagging); when complete merge them into trunk, and branch a release which went through user-acceptance and production environments. While we still had the odd issue with branches conflicting, we did however we managed to resolve a lot of the other issues we had.

Baking in the tooling and measuring systems in is only part of the journey; there is also reporting on your non-functional requirements; Establishing knowledge around how the systems behave so you can work on alerting for when systems start misbehaving.

Organisational Skills

- As a Solution lead on a range of systems; Maintaining close relationships with clients, growing the professional relationship with a client to a trusted advisor which enables you recommend development, and process changes in conjunction with a client's business needs. Coordinating work for my team; from requirements gathering, specification, estimation to delivery. Applying aspects of Dev-ops ideas; and Agile principals to gain benefits that work for both the team and the client
- As a mentor, or technical expert on a range of systems; Guiding team members towards developing the skills to formalize, specify, estimate, and deliver against a business requirement. Sometimes this involves guidance from a business process, system architectural or lower level implementation detail. I believe in establishing a zero-blame culture, I by default take the position that there are no 'daft questions' and everything is a learning opportunity irrespective of experience, familiarity or tenure in an organisation or system. I receive consistent positive feedback form my line managers about how I work with my more junior co-workers and help grow them into well rounded professionals.
- As an experienced software developer; 15 or so years is a lot of time to summarise; irrespective I have a lot of experience working independently, in teams, managing my own backlogs, managing my teams, and working against backlogs managed by other people. I spend a lot of my own time reading about ideas, code and systems as well as working on my own projects (see my github links above) which helps solidify ideas.

Experience

Intergen: 2007 - Present

- Type of Business: Professional Services, Information Technology Consultancy
- Role: Software developer, Application support consultant, Problem Solver

As a problem solver, I work closely with a client's subject matter experts, and stakeholders to shake out formal requirements, estimate, and execute the requirements until completion. Depending on the requirements I will either lead a team towards completion, execute completion myself. My role varies between backlog grooming, unblocking problems, implementation tasks, incident management, and resolution.

Achievements

- Championing using Azure DevOps for continuous testing and delivery In the place of nothing *azdo* allows for a simple yet powerful way to achieve a high level of automation around testing and delivering of code changes into client environments. Over the past few years I have pushed the support side of the business to use more of the testing and releasing functionality provided in *azdo*; in doing so I have been able to migrate a number of projects to use *azdo* for continuous-testing and releasing.
- Championing documentation An opportunity has been apparent for some time for somebody to pick up around project documentation; so I decided to champion that and develop a set of documentation templates to replace our existing wiki tooling that is at the end of its life. Over the past year or so working with management, delivery, and support teams I have put together a set of templates which provide a skeleton for project and support teams to start from, as we are formalising on Azure DevOps Wiki pages for project documentation the templates are designed to be imported into a *azdo* Wiki site where the team can start filing them out from there using the templates as a guideline for the sort of information that is needed and encouraging them to write more about their solution.
- Source control guidance (Team Foundation Server) I identified an opportunity to improve the process around the safe and effective use of branching strategies; once identified I spent some time putting together guidance around practices that were common elsewhere in the software-development world; we circulated these practices in our team for a number of months responding to the odd adjustment; eventually bringing them to the wider organisation as a set of minimum practices and expectations that we as a dedicated product support organisation require implemented on a project.
- Source control guidance (GIT) With the organisation as a whole moving to projects source controlled via GIT, I developed guidance around some very common branching strategies and the sort of projects that best suit the common strategies.
- Source control migration guidance I saw an opportunity to put together some guidance around migrating various combinations of team foundation server (on-prem or *azdo*) to GIT; depending on the requirements (e.g. tfs-source history) it can be simple or require some third party tools and command line scripting. For a while I was

a subject matter expert and assisted several projects with the migration from TFS into GIT.

Work in Progress

- Working towards developing my Azure DevOps expertise In my functional area we spend more time making changes to systems running in production; sometimes the production environment is an on-prem server / or virtual machine, other times it is infrastructure as a service or platform as a service. In my quest for championing continuous delivery as default; I am spending some time learning to appreciate some of the more advanced aspects of Azure Release Manager Templates, where you specify your environment as configuration and use *azdo* release tasks to deploy or update your environment

Responsibilities

Solution lead, Property and Land Management System

C#, ASP.NET Web-forms, SQL, Office COM Automation, Visual Studio Team Systems, Branching methodology.

Multi-tenant district valuation and role system used by a number of agencies to record information against land and properties. Parts were website driven but also has a lot of job automation where it would pick up data form and export to a number of different places.

As a Lead Developer

- I managed most day to day tasks including coordinating the work amongst my team; code reviews; working closely with internal owners and the client; managing expectations around completion, estimates and overrun - in between the management I also wrote my fair share of code.
- *Acting as a trusted advisor; I worked closely with the client's subject matter experts to put forward ways to improve various aspects of the solution.
- *Working with client subject matter experts we formalize, specify, estimate, and deliver a wide range of enhancements to the solution.
- Acting as a technical solution expert; working closely with my team members to guide them around the solution working with them to implement specified changes.
- Responding to production incidents; incident management, troubleshooting and incident resolution

Highlights

- Large old codebases always have their own set of challenges; highlights include 'imparting the secret knowledge of how old systems are written' to the more junior team members and watching them 'get it' (mainly because I am old enough to have written code like that).
- Originally the project used an old version of TFS (Team Foundation Server); had no branching methodology; and a large product backlog. Once we got branching under control we moved to managing the backlog in Azure Dev-Ops and eventually migrated form on prem tfs to GIT on *azdo* (including all 14 years of history).

Solution lead, on a Multi-Tenant Finance Web Application

ASP.NET, MVC, SQL, SOAP over WCF Web Services

Multi-tenant finance brokering system; for managing finance contract lifecycles.

As a Lead Developer

- Work with client subject matter expert, formalizing, specifying estimating and delivering work on the solution.
- Where there is a large amount of work I have worked in a small team to deliver the work after a formal requirements phase

Contributor, Subject Matter Expert on Patient Healthcare System.

XAML, Windows Universal, ASP.NET MVC, ASP.NET WebAPI, SQL, Visual Studio Team Systems Continuous Integration, and Release Management pipelining (vsts devops)

Highly interactive patient healthcare system; replacing a paper/electronic system with a purely electronic system that all patient healthcare practitioners used.

As a Team Member

- I have worked as a part of a large agile based team on sprints delivering against a product backlog
- My particular focus was on preparing the solution for an eventual support and maintenance phase by a dedicated team; alongside delivering works in the product backlog my focus on preparing documentation and processors.
- Over my time I ended up touching most aspects of the system including release scripting, front end XAML, back end services.

Application support consultant

C#, ASP-MVC, dotnet-core, SQL, Azure DevOps

The work I end up doing can vairy a bit wildly but can be summed up as.

- As a Senior developer I am a lead support agent on several systems for a number of clients, paying particular attention to the ones I own on a daily basis.
- As an incident manager; sitting in between technical resolve team and the client on high criticality incidents. My main job is to act as a conduit for comms between the technical resolve team and the client; where necessary unblock the technical-resolve team.
- As an incident resolver; work on the pointy end to resolve a production incident with a client; usually by acting in the technical-resolve team you are a subject matter expert on the affected system or less common the impacted technologies if the sme's are running into resolution issues.
- For the systems I own on a daily basis I do the day-to-day incident resolution and development; again the work can vary wildly form one system to another; but generally I do a lot of estimating, estimate-review, code review, knowledge sharing, and software development.

- Being one of the old hands in the organisation at one point in time or another I have had my hand in a lot of systems; had my fair share of critical incident resolution and management. Highlights would have to include the feeling you get when you are deep into resolving a critical incident; and finally get to a resolution.

Kognition Consulting Limited: 2005 - 2007

- Type of Business : Professional Services, Information Technology Consultancy
- Role : Software developer, Solution Implementer

I Started off as an Implementer; given a specification, completed them under guidance from more senior team members. Progressed towards managing small projects; taking a formal specification, estimating and the delivering of work.

Responsibilities

- Windows mobile subject matter expert; particularly cross platform concerns.
- Worked intimately with a field services organisation; their stakeholders and subject matter experts to estimate and deliver a wide range of specified enhancements to existing large cross platform systems.
- Worked at an Implementer level with several different clients; their stake holders and subject matter experts as a part of a larger team to deliver a number of systems and enhancements.

Education

- Date: 2006
- Institution: Otago Polytechnic, Dunedin, New Zealand
- Formal Qualification: Bachelor of Information Technology (BIT)
- Subjects and Skills: Systems Analysis, Project Management, Quality Assurance and Software Testing, Embedded Systems, Mobile, Desktop and Web Development

References

Available on Request.