

# Profile of Michael Welch

---

## Summary

---

I have 19 years of development and IT experience and 3 years of formal leadership experience. I bring a unique mix of skills as I've contributed, led, mentored and taught. I have an educational background in engineering, computer science and business.

My passion is developing clean quality code and mentoring others to do the same. I stay sharp by doing independent reading, contributing to projects on GitHub, creating my own personal study projects, taking university courses, and facilitating study groups.

I have 3 years of agile experience.

## Career

---

### Employment: Johnson Controls

*Lead Staff Engineer (and previous titles): 2002 - Present*

I've worked on many releases and features of the Metasys building automation system and P2000 security & fire systems over 13 years. The majority of that experience involved the following technologies: C#, Java, Sql Server, web services.

### Highlights

- Implementation of binary serialization framework in Java and C#
- Introducing unit testing, refactoring, and clean coding through mentoring, code reviews, and study groups
- A language for expressing constraints and a compiler for generating code to enforce them
- State machine engine
- Team member on MMS (a distributed communication protocol and related set of application services)
- Team lead for reusable controls for Graphics+ project
- The first to use distributed source control on a project/team
- Recruited and managed a highly skilled team that incorporated the best of unit and integration testing, continuous builds/integrations, good code review practices and auto-deployment.

### Projects

#### Global Data Platform

*May 2015 - Present*

I recently started a development position on our global data platform team. Working with Microsoft Azure stack to develop "connected" chillers solution - allowing for the collection and analysis of customer data.

The component I'm working on is a "protocol gateway" used to connect Johnson Controls devices to a Microsoft EventHub. This protocol gateway converts messages from the proprietary JCI MMS protocol to AMQP.

*Key concerns:* scalability (how many messages can we handle), efficiency (how quickly can I process messages and reply with an acknowledgement to sending device), error conditions (what happens when EventHubs throttle or time out, what happens if we are sent a message larger than EventHubs can handle).

## **P2000 User Interface**

*October 2011 - May 2015*

I served as the technical lead of the P2000 user interface team. The team consisted primarily of consultants all of us new to the P2000 product. My role included high level design, mentoring a team of 7, implementation, and code reviews. Technologies used were HTML5, JavaScript, and ASP.Net MVC. A year after starting this team I was promoted to Engineering Manager of the same team.

## **Graphics+ Feature**

*October 2009 - October 2011*

For two years I served as the technical lead for our Graphics+ feature of our building management system. The tool allows a customer to build “smart” graphics to monitor and command their building control system. Required the creation of 300 custom WPF/Silverlight controls. These controls also needed to work seamlessly in a layout tool (think Blend) and allow these controls to be bound to real-time information (thermostats, damper positions, etc.). The team grew to approximately 25 people at its largest.

Key problems solved: \* Design of custom URIs to allow controls to be bound to objects from different real-time systems. \* The packaging of graphical controls so that they could be used in an editor as well as runtime (as well as be used outside of the editor) \* Embedding our Silverlight runtime in ASP.Net and also a Java app. \* Fast messaging to meeting CTQs (critical to quality). Change of values and alarms must be presented in the user interface within 5 seconds of the event being detected by the system.

## **MMS and BACnet Encoding**

*2005 - 2009*

Around 2005 a team was tasked with allowing our distributed controls systems to scale up to support a larger number of devices and to be more responsive. A key component of this was the development of a proprietary communication protocol that allowed for more efficient communication between devices. Rather than every task sending a message whenever it wanted, communication was funneled through a central transport where they could be bundled up and sent together. Required the development of efficient encoding, bundling and routing mechanisms. Another key components was to develop an application services framework which simplified the development of “tasks”.

Key problems solved: \* I worked on a gateway between our managed (C#) tasks and our main native application services framework. \* Developed `BACnetReader` and `BACnetWriter` used to develop encoders/decoders for BACnet binary schemas. Frameworks were written by me in C# and Java. (For reference, a [Haskell version](#) was written by me recently, for fun - on my own time, and is available on GitHub.) \* Developed a C# BACnet schema validation framework. It is used in unit testing to make sure C# encoders generate payloads that conform to their respective schema. The framework also ensures that payloads can be “round-tripped”. This framework could be consumed by both Java and C# developers.

## **Employment: SysLogic**

*Senior Consultant: 1999–2002*

I was a Senior Consultant. I developed in Visual Basic, C# and Sql Server for various clients.

## Employment: Procter & Gamble

*Systems Analyst: 1992–1997*

I was a Systems Analyst. My responsibilities included requirements analysis, project management, system support and some development for ERP type systems: production planning, material purchase planning, and warehouse utilization planning.

## Management Experience

---

*October 2012 - May 2015*

I was promoted to engineering manager a year after forming the P2000 Web UI team. My biggest contribution as manager was to recruit the best employees from within JCI. They've been given considerable freedom and have exceeded my expectations.

In my role I needed to pick a development practice (I chose scrum), mentor the team as we learned scrum together, assemble best practices from the accumulated knowledge of all consultants, pick a tool chain, hire skilled staff, etc.

## Teaching Experience

---

### Milwaukee School of Engineering

*Adjunct Assistant Professor: 2007, 2009*

I have taught two courses at Milwaukee School of Engineering: Data Structures and Software Development I.

## Education

---

- M.S. Computer Science, University of Wisconsin - Milwaukee
- M.S. Business, University of Wisconsin - Madison
- B.S. Electrical Engineering, University of Wisconsin - Madison

## On the Web

---

My blog is at [loominate.net](http://loominate.net). There I occasionally write about software development, computer science and math.

I have many repositories on [GitHub](https://github.com) and [Bitbucket](https://bitbucket.org). Favorites include [mbasic99](#) (a compiler for TI BASIC), [tibasic](#) (A port of mbasic99 to Apple's Swift language), [brainmess](#) (a simple interpreter useful for refactoring study groups), [parser.js](#) (a functional parser library in JavaScript) and [loominate.net](http://loominate.net) (my blog).

I've asked and answered a handful of questions on [Stackoverflow](https://stackoverflow.com)

## Hobbies

---

In the last couple of years I've taken an interest in physical fitness: In the last 3 years I've completed my first three half-marathons, a Rugged Maniac and a Tough Mudder. I'm currently training for my first marathon in the fall of 2015.

I enjoy keeping up with tech industry and read several blogs related to C#, Swift, Haskell, functional programming, and Apple.

I enjoy studying math (for example, [abstract algebra](#)) and learning new programming languages (Haskell, JavaScript and Swift being the most recent).

## Favorite Texts

---

- [Refactoring](#) by Martin Fowler (Chapters 1 and 3 by themselves are worth it.)
- [Test Driven Development](#) by Kent Beck
- [Clean Coder](#) by Robert Martin
- [Clean Code](#) by Robert Martin