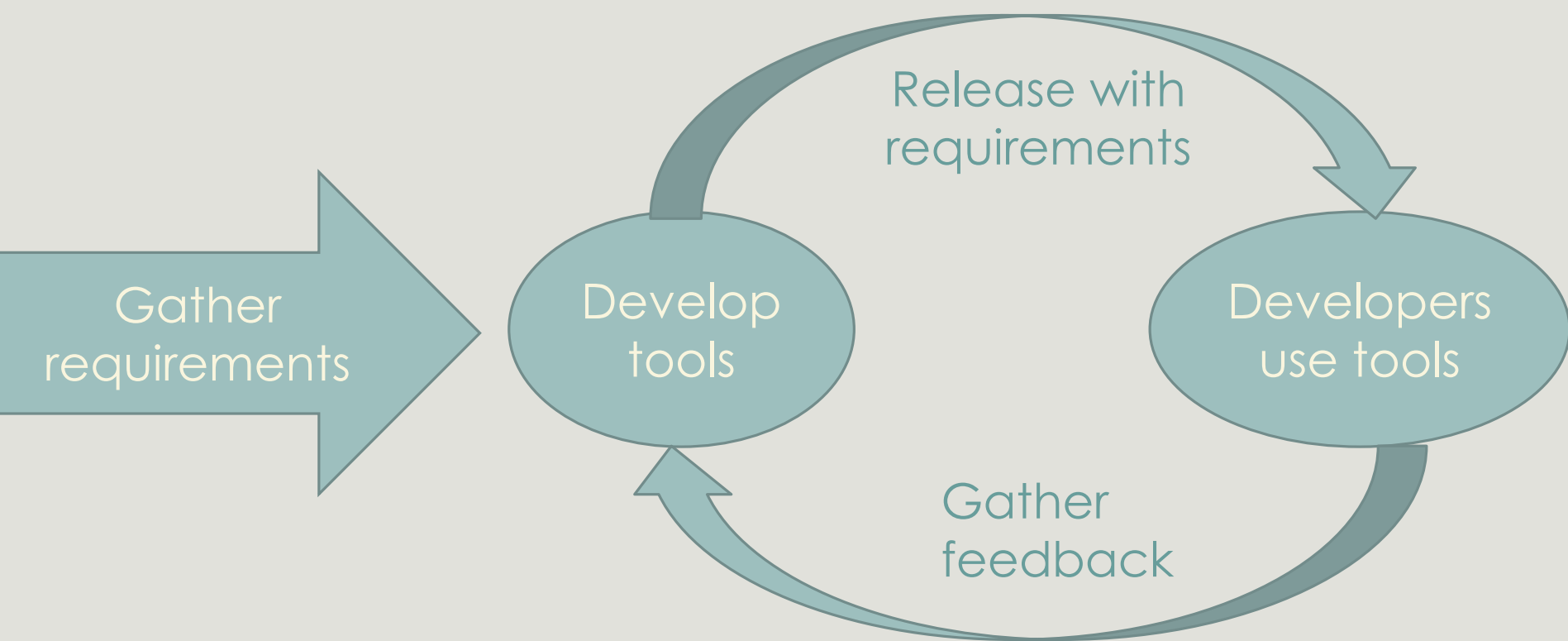


Duality

Previously coined by Git and others, Developer Driven Development allows development and work to be guided not by SCRUM/Agile or other arbitrary rules but by those doing the development. Simply, given requirements get work done. The LTE RRC team at Qualcomm wholeheartedly embraces this model and allowed myself and my coworkers the freedom to complete our tasks during the work day without having to deal with the needless overhead of managing a SCRUM board.

At the same time, projects during my time at Qualcomm grew out of this workflow. In order for developers to maintain freedom, they needed the ability to automate the monotony of the everyday. With this, I was able to work on tools for my team with requirements driven by the developer's needs.

My DDD Workflow



From UCSD

Not surprisingly, my education actually prepared me for the tasks I faced during my time at Qualcomm. In one way or another, I feel that I used something from each course I've taken to help me succeed. CSE 110 helped with designing each of my projects to allow extensibility. CSE 20/21/100/101 helped me analyze my solutions and determine if they were optimal or if I would run into trouble down the line. CSE 105/140/141 helped me better understand the architecture side and design a clever system by modeling it after models of computation that I am already familiar with. CSE 11/12/15L/30 helped me understand the necessity of documentation and clean code, as well as helping me to debug quickly. On more than one occasion, I found bugs during meetings just by seeing a few signs that things were not right and was able to fix them just as quickly. All of these classes worth of experience, coupled with my ability to work both independently and in a team setting enabled me to work to the very best I am capable of.

UE Capability Validation

My first project at Qualcomm was to create a two stage validation for radio configurations on the LTE call stack. The first validation is against the 3GPP(LTE Governing body) Technical Specifications to ensure that Qualcomm modems don't communicate in ways that are out of spec. The second validation is against the actual configuration that occurred to ensure it occurred as expected. This project was expected to take the entire 13 weeks and was the project I was selected for. I had a complete build in 4 days, and it was released 2 weeks from my start date.

	Band 1	BandwidthClassDL-r10	BandwidthClassUL-r10	Band 2	BandwidthClassDL-r10	BandwidthClassUL-r10	Band 3	BandwidthClassDL-r10	BandwidthClassUL-r10	SpecValidated
1	42	a	a							yes
2	41	a	a							yes
3	40	a	a							yes
4	38	a	a							yes
5	39	a	a							yes
6	42	a	a	42	a					yes
7	42	c	a	41	c					yes
8	42	c		41	c	a				yes
9	42	a	a	41	d					no
10	42	a		41	d	a				no
11	42	a	a	41	a					yes
12	42	a				a				yes
13	42	c	a	41	a					yes

Sample Validation output

ISF Parser

My second project at Qualcomm was to create a parser for log files to extract all of the debug messages, filter, and categorize them so they would be more useful to developers. After developing this tool per my DDD workflow, teams other than the one I was assigned began to request integration of this tool with their team. As such, it is designed to be fully extensible and configurable, so that any team/developer could adopt it and modify it to fit their needs without ever having to touch source code.

Crash Triage Automation

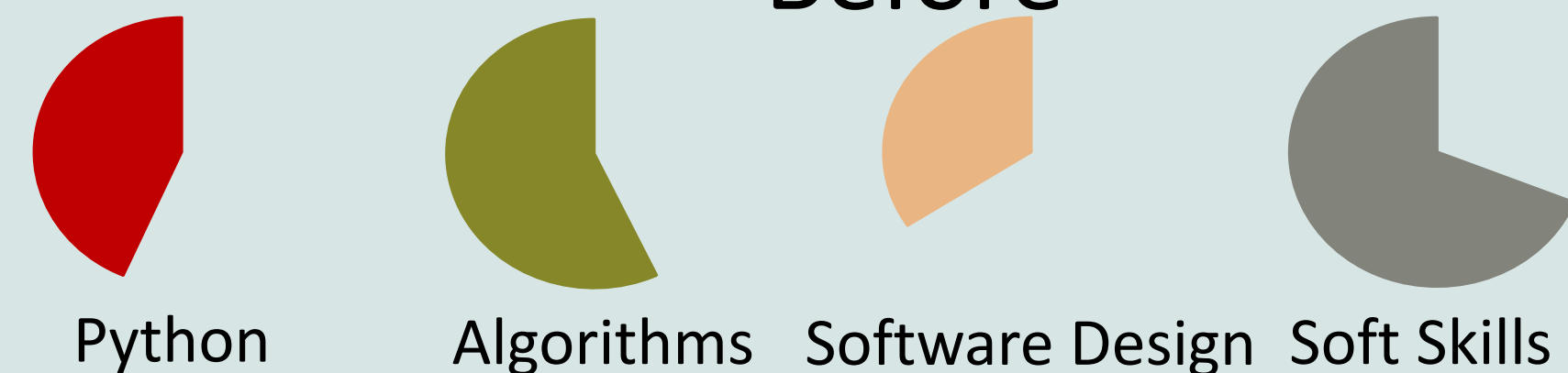
My third project at Qualcomm was to integrate my team (RRC) with a framework that was already developed for automating crash analysis. This automation on my part required getting up to speed on this framework and integrating my team's work so that developer's could save time both in their own use, and the use of others. With automation in place on a server, anyone with questions about the crash can be redirected to check the server first. Thus, DDD is preserved and developer's can work freely without interruption.

Automation Tools

My other projects at Qualcomm were a series of updates to other tools already in use. One tool was updated to generate bit masks automatically based on selections, versus working out the mask by hand and generating the file. Another tool validates string encodings of phone capabilities and creates an estimate for how many bits/bytes those capabilities will take when they are encoded and transmitted over the air



Before



After

