

Trent Forkert
 CS 460
 Exam II

1. Overview Table

Roles	Developer, Database manager, Git curator, System architect
Tools	Eclipse, psql, ssh, git, Google Docs
Major Assigned Tasks	<ul style="list-style-type: none"> ● Become Sugar Expert ● Create and maintain Git Repo ● Create and maintain postgres server ● Write business rules for CS degree requirements ● Create requirement selection ui
Major Completed Tasks	<ul style="list-style-type: none"> ● Become Sugar Expert ● Create and maintain Git Repo ● Create and maintain postgres server ● Write business rules for CS degree requirements

2. Major Tasks

Name	Description	Begin	End
Sugar Expert	Understand how Prof Wolfe's code uses sugar, be able to help teammates and solve problems.	Week 3	Ongoing - continuous task
Git curator	Create and manage the team's git repository. Help teammates to use it correctly and solve problems with it that arise.	Week 1	Ongoing - continuous task
Postgres maintainer	Create and manage a postgres database for Connor's BingoSheet.	Late February	Ongoing - continuous task

Business Rules	Write the code behind "Business Rules", degree requirements that cannot be adequately represented by the Registrar's database	Week 3	Ongoing - some business rules are too rigid for practical use
Requirement Selection UI		Late March	Ongoing - nearly finished

3. A selection of SWEBOK Reviews

3. Software Construction

This Knowledge Area describes the actual implementation of the software system, in accordance to the requirements and design. However, this is more than just programming, but includes some debugging and testing.

Large parts of this knowledge area are dedicated to constructing software in such a way that the task naturally makes itself easier and more fault-resistant. This is where topics such as standards conformance, unit tests are of incredible use and importance.

This KA also describes what I consider good basic programming skills, namely minimizing complexity and code reuse. To me, these parts of the knowledge area seem reminiscent of the UNIX Philosophy, where you make one piece of code do one simple thing well, and build up complexity by combining smaller pieces of code.

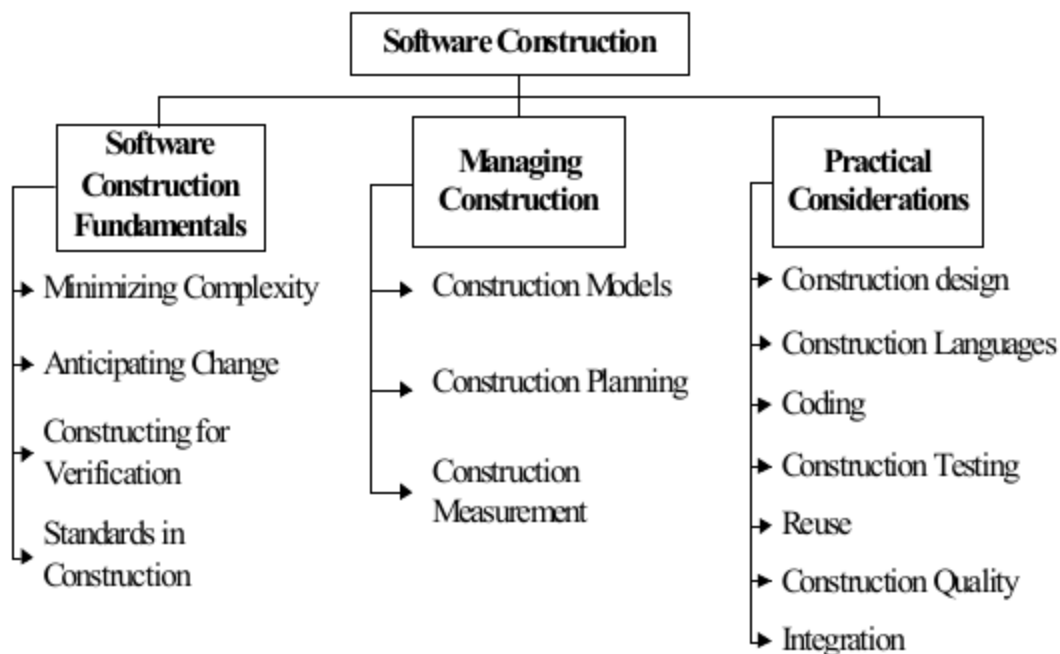


Figure 1. Breakdown of topics for the Software Construction KA.

A large part of my roles for this semester has been writing (or constructing) software. So this particular KA is fairly directly related to my roles.

4. Software Testing

This knowledge area describes testing, as well as techniques and processes for testing software systems. The Software Testing Knowledge Area stresses the fact that tests must be carefully selected, so that a relatively small number of tests have a significant impact and meaningful results. Testing also explicitly happens by actually executing code.

This KA describes different goals of testing, such as requirements testing and

correctness testing, where the software is tested to see if it meets the customers requirements and behaves correctly, respectively. Also described is stress testing, the process of seeing how the software performs under very high load. This will have direct application to our project.

Additionally, a large number of techniques for testing are described, including fault-based techniques and usage-based techniques.

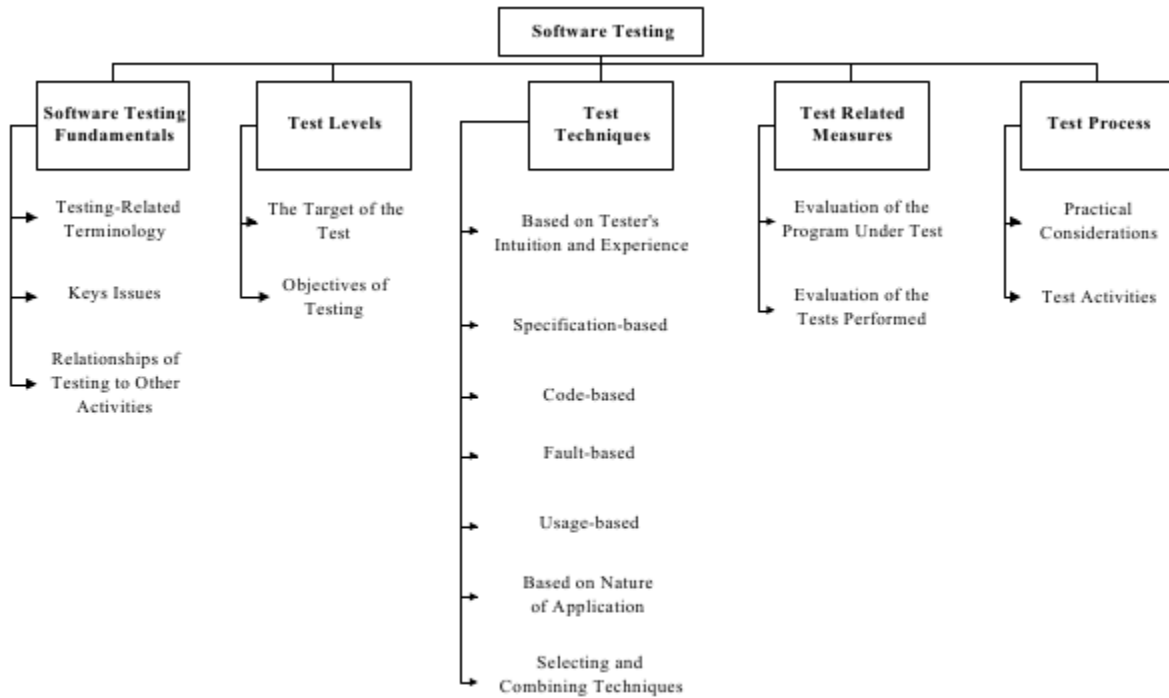


Figure 1 Breakdown of topics for the Software Testing KA

Alongside developing software, I also have to test all my code to ensure that it performs as expected.

5. Software Maintenance

This knowledge area describes software maintenance, its processes and techniques. This includes planning for maintenance prior to release, as well as fixing problems and providing support after release. Maintenance includes reevaluating the software to make sure it still meets requirements, or indeed whether or not those requirements need updating later in the software's life cycle.

Part of this knowledge area is concerned with the categorization of different types of maintenance, with four main categories (preventive, perfective, corrective, and adaptive). Also discussed are the cost implications of continued maintenance, and the cycle of the software maintenance process.

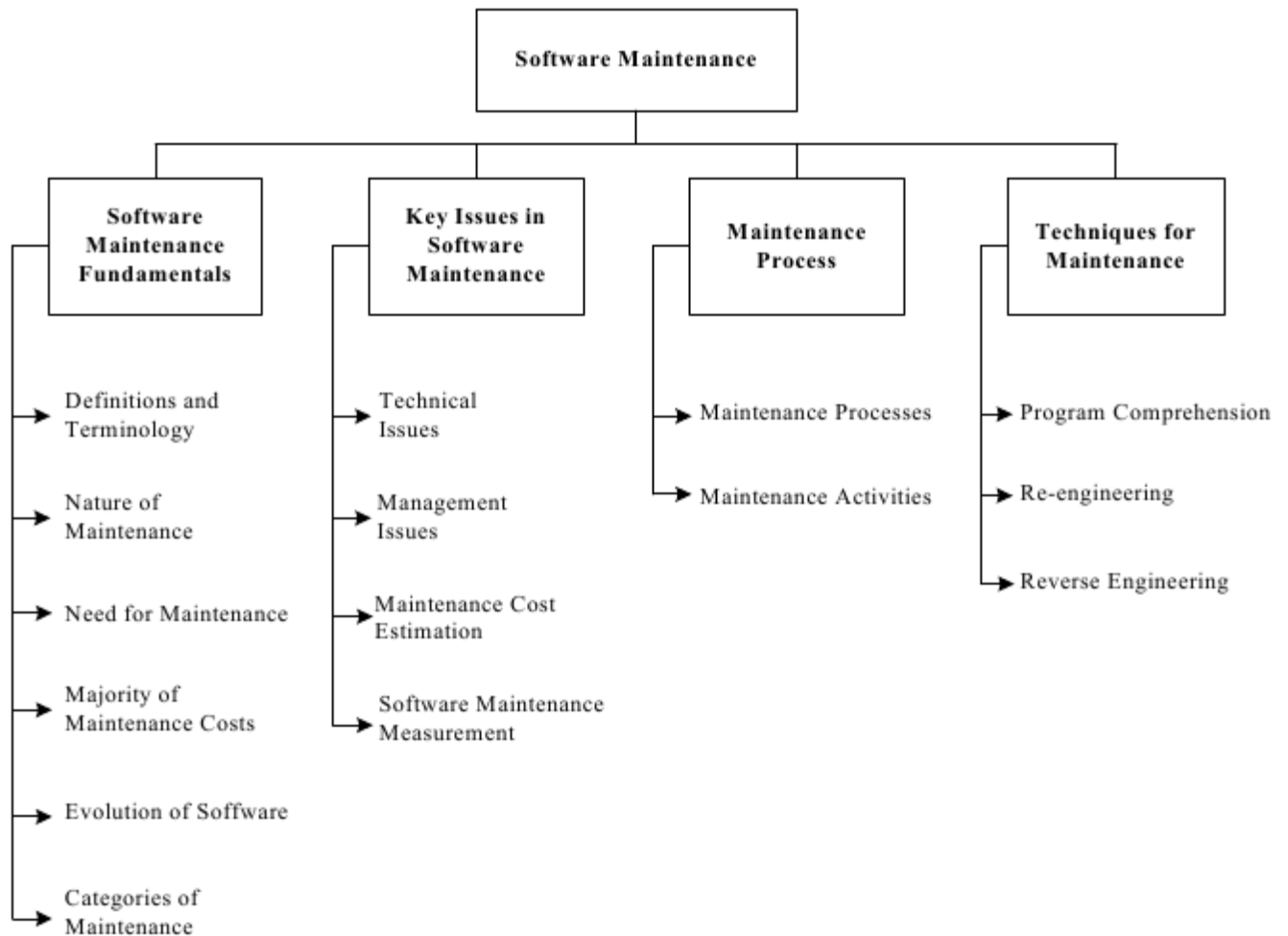


Figure 1 Breakdown of topics for the Software Maintenance KA

On occasion, I have to maintain features I have developed, when they need additional functionality or if a problem arises that I need to fix.

4. The following table contains all progress reports from this semester I still have record of.

Week	Completed	Planned
Feb 1 - Feb 8	<ul style="list-style-type: none"> ● Fixed Git/Eclipse project files problem ● Did 3 slides for presentation ● Further updates to Use Case as per Prof S's feedback ● Wrote up a small table in my notes giving each Use-Case/Construction area a priority. ● Updated my CMap 	<ul style="list-style-type: none"> ● Import the project files from Marat/Prof Wolfe into Eclipse ● Investigate the Scheduler class from Prof Wolfe's code. ● [Optional] Figure out a clean way to reference the Project's dependencies without forcing binary blobs into the Git repository. ● Get project onto the Git repo so we can all synchronize our changes ● Second attempt at Friday meeting, now at a different time.
Feb 8 - Feb 15	<ul style="list-style-type: none"> ● Imported the project files from Marat/Prof Wolfe into Eclipse ● Investigated the Scheduler class from Prof Wolfe's code. ● Put Code as it existed on my machine onto Git. <ul style="list-style-type: none"> ○ Left JAR files in-repository (for now?) ● Met on Friday <ul style="list-style-type: none"> ○ Found a plugin and set up Eclipse for visual 	<ul style="list-style-type: none"> ● Figure out how to get a list of courses that satisfy GenEd reqs ● Figure out how to get a list of courses that satisfy Adv. Com req. ● Work toward becoming an expert on Sugar and how we are using it ● Put updated code on Git if and when I receive it from Prof S.

	<p>access to the database</p> <ul style="list-style-type: none"> o Helped Connor and Alek get Eclipse set up for Git 	
Feb 15 - Feb 22	<ul style="list-style-type: none"> ● Determined that our database does not have Gen Ed info ● Wrote helper class to get list of Adv. Com. courses ● Wrote document on how we are using Sugar ● Determined newly received code is not something to push to Git 	<ul style="list-style-type: none"> ● Update my slides for Presentation II ● Write other business rules like Adv. Com. <ul style="list-style-type: none"> o Lab Sciences o Coordinate with Connor o Investigate creating a common API for business rules
Feb 22 - Feb 29	<ul style="list-style-type: none"> ● Wrote preliminary Lab Sciences business rule ● Helped other get code on Git ● Updated slides for Presentation II 	<ul style="list-style-type: none"> ● Further update slides for belated Presentation II ● Clean up and test existing business rules when ODS is fully back online ● Set up our postgres/tomcat server once Prof S has it ready <ul style="list-style-type: none"> o Make it meet group's demands ● Work with Connor, Yei Sol and Marat to provide business rule satisfiers as needed
Feb 29 - Mar 14	<ul style="list-style-type: none"> ● Further update slides for belated 	<ul style="list-style-type: none"> ● Set up Postgres on softbase (the server

	<p>Presentation II</p> <ul style="list-style-type: none"> ● Clean up and test existing business rules when ODS is fully back online ● Set up our postgre/tomcat server once Prof S has it ready <ul style="list-style-type: none"> ○ Make it meet group's demands ● Work with Connor, Yei Sol and Marat to provide business rule satisfiers as needed 	<p>Prof S is giving me access to)</p> <ul style="list-style-type: none"> ● Test and deploy my Tomcat solution ● Work with group to get their code and systems ported over to the new Tomcat and postgres configs ● Finish up existing business rules, pushing them to git ● Take another crack at a business rule for Gen Ed requirements
<p>Mar 14 - Mar 21</p>	<ul style="list-style-type: none"> ● Set up postgres on softbase ● Figured out a tentative tomcat solution, still needs more testing ● Gave Connor the needed postgres connection details ● Finished and committed AdvCom and LabScience business rules ● Resolved to hard-code GenEd reqs for the time being 	<ul style="list-style-type: none"> ● Finish the faked GenEd business rules ● Schedule GUI: choose which requirements to schedule for on a BingoSheet <ul style="list-style-type: none"> ○ Integrate with YeiSol's tabbed presentation (mostly done) and Alek's schedule display (new this week) ● Finish testing and deploy my tomcat set up ● Update business rules as needed if teammates require additional functionality ● Assist team with postgres/git if

		needed
Mar 21 - Mar 28	<ul style="list-style-type: none"> ● Finished the fake GenEd business rules ● Large chunk of Schedule GUI finished <ul style="list-style-type: none"> ○ The combobox is done, and accepts inputs ● Pushed Tomcat setup to git 	<ul style="list-style-type: none"> ● Finish my pieces of the Schedule GUI <ul style="list-style-type: none"> ○ Integrate with YeiSol and Alek's work ● Git workflow adjustment and maintenance ● Thoroughly debug my work
Mar 28 - Apr 4	<ul style="list-style-type: none"> ● Schedule requirement selector ui mostly complete ● Transitioned team to new git repo, workflow <ul style="list-style-type: none"> ○ Added basecamp integration to new repo 	<ul style="list-style-type: none"> ● Tweak requirement selector UI <ul style="list-style-type: none"> ○ List requirements based on a given student's bingo sheet ○ Functionality tweaks as suggested by Prof S. ○ Track down Gen Ed bugs ● Pass selector outputs to Sugar, Sugar's output to Alek ● Make a JSP to handle both creating and displaying a schedule <ul style="list-style-type: none"> ○ Referencing Alek's and my JSPs
Apr 4 - Apr 11	<ul style="list-style-type: none"> ● Semester selection box is now separate from the requirement 	<ul style="list-style-type: none"> ● Send user provided time restraints (from iCal), min/max credit hours and

	<p>choosers</p> <ul style="list-style-type: none">● Gen Eds partly fixed, some rules still need updating● Requirement Chooser now generates text file with choices formatted for Sugar● Personal CMap updated (layout, some new content)	<p>chosen requirement satisfiers to Sugar, provide a structure describing the result to the session object</p> <ul style="list-style-type: none">○ Use unique temporary filenames based on student ids○ Rewrite existing Sugar calls to generalize them as needed <ul style="list-style-type: none">● Integrate into Connor's APIs
--	--	---