Module 1 - Supplemental Resources

Processes and Practices

This article examines Rational Design processes, and how they work, and why you should fake them (if they are not actually executed). This is additional information to the course, and not a mandatory reading.

PARNAS, DL. "A Rational Design Process: How and Why to Fake It." 1986. <http://www.ics.uci.edu/~taylor/classes/121/IEEE86_Parnas_Clement.pdf>

The Wikipedia article that outlines software engineering processes. Just for reference if you wanted any extra clarification on the subject.

"Software development process - Wikipedia, the free encyclopedia." 2011. 21 Jun. 2016 <https://en.wikipedia.org/wiki/Software_development_process>

An article that outlines some practices that can make you a better programmer. Not in the code sense, but more in the team member sense. An interesting read whether you are a programmer or not.

"The Ten Commandments of Egoless Programming - Coding Horror." 2014. 21 Jun. 2016 <https://blog.codinghorror.com/the-ten-commandments-of-egoless-programming/>

A really neat TedTalk in which Tom Wijec shows how you different approaches for evaluating problems in your company. He uses the example of how to make toast, and demonstrates different ways of breaking down the process. This relates to this module because processes are all about breaking down the overall goal in a systematic way. This provides a great reason for why we have included the subjects in this module.

"Tom Wujec: Got a wicked problem? First, tell me how you make toast ..." 2015. 21 Jun. 2016 <https://www.youtube.com/watch?v=_vS_b7cJn2A>

Software Engineering Activities

This is the formal document that outlines the software engineering activities and how they fit together in processes. Essentially, this document outlines how one could create their own process. Our lecture is based off this and simplified. The IEEE is the Institute of Electrical and Electronics Engineers. The official link may not work if you do not have access to the IEEE library, so we have provided an additional link that should work.

"IEEE Xplore Abstract - IEEE Standard for Developing a Software ..." 2012. 21 Jun. 2016 <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=1665059>

<http://arantxa.ii.uam.es/~sacuna/is1/normas/IEEE_Std_1074_1997.pdf>

Module 2 - Supplemental Resources

Linear Models

This is a super relevant and interesting read for this module. It explains how the Waterfall model was a misunderstanding and how it was never actually intended to be used. This article also highlights how some of the findings in the original paper on the Waterfall model (which was written way before the Agile Manifesto) actually align with Agile philosophy.

"Why Waterfall was a big misunderstanding from the beginning ..." 2012. 21 Jun. 2016 <https://pragtob.wordpress.com/2012/03/02/why-waterfall-was-a-big-misunderstanding-from-the-beginning-reading-the-original-paper/>

This paper shows a timeline of some software and project management methods. It's a long article and it may not be worth reading the entire thing, but has a visual timeline which is useful and interesting.

Rico, DF. "SHORT HISTORY OF SOFTWARE METHODS by David F. Rico This ..." 2011. <http://ww.davidfrico.com/rico04e.pdf>

Spiral Models

A good paper that explains the Spiral Model in a detailed manner. Worth reading if you are interested in pursuing the Spiral Model.

"A Spiral Model of Software Development and Enhancement." 2015. 21 Jun. 2016 <http://csse.usc.edu/TECHRPTS/1988/usccse88-500/usccse88-500.pdf>

Another detailed explanation of Spiral Model. Does a good job of explaining all the invariants in the Spiral Model.

Boehm, B. "Spiral Development: Experience, Principles, and Refinements." 2000. <http://www.sei.cmu.edu/reports/00sr008.pdf>

Unified Process

This link gives you an excerpt of the book Unified Process Explained. These chapters explain some of the history, background knowledge, and execution of Unified Process.

"Overview of the Unified Process | Introduction | InformIT." 2007. 21 Jun. 2016 <http://www.informit.com/articles/article.aspx?p=24671>

A basic explanation of Use Case Diagrams. The link includes some examples of Use Case Diagrams as well.

"Use case diagrams are UML diagrams describing units of useful ..." 2010. 21 Jun. 2016 <http://www.uml-diagrams.org/use-case-diagrams.html>

A basic explanation of UML Diagrams. Also includes example diagrams.

"UML Class and Object Diagrams Overview - common types of class ..." 2011. 21 Jun. 2016 <http://www.uml-diagrams.org/class-diagrams-overview.html>

Prototyping

A good overview of some of the different types of prototyping. Doesn't go into much detial, but provides a good explanation.

"SDLC - Software Prototype Model - TutorialsPoint." 2013. 7 Jul. 2016 <http://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm>

A really interesting TedTalk that explains the value of prototyping and process. I highly recommend that you give this a watch.

"Tom Wujec: Build a tower, build a team - YouTube." 2010. 21 Jun. 2016 <https://www.youtube.com/watch?v=H0_yKBitO8M>

Continuous Delivery

A really good site that explains Continuous Delivery and it's principles and foundation. It also provides case studies and evidence supporting Continuous Delivery.

"What is Continuous Delivery? - Continuous Delivery." 2010. 7 Jul. 2016 <https://continuousdelivery.com/>

In this module, you learned that Microsoft continuously integrates their code via the Microsoft Daily Build. This resource explain that process in more detail.

Cusumano, MA. "How Microsoft builds software - ACM Digital Library." 1997. <http://dl.acm.org/citation.cfm?id=255656.255698>

This article explains what a Daily Build is an some of the advantages of using this method. It's a relatively short article and a good read if you are interested in learning more.

"Daily Build and Smoke Test - Steve McConnell." 2003. 21 Jun. 2016 <http://www.stevemcconnell.com/ieeesoftware/bp04.htm>

Module 3 - Supplemental Resources

Using Agile with Process Models

This is the official resource for the Agile Manifesto. It simply outlines the values, principles, and history of the Agile Manifesto.

"Agile Manifesto." 2003. 21 Jun. 2016 <http://www.agilemanifesto.org/>

A book that highlights that the major issue with software development is people. The book provides solutions to dealing with the people in software projects. An interesting read.

"Peopleware: Productive Projects and Teams (Second Edition): Tom ..." 2016. 21 Jun. 2016 <https://www.amazon.com/Peopleware-Productive-Projects-Teams-Second/dp/0932633439>

This post provides a good explanation of traditional models and Agile methods. It also talks about some challenges you may face when transitioning from a traditional model to Agile. It provides a good overview for the topics explored in this course.

"Traditional and Agile Methods: An Interpretation - Scrum Alliance." 2013. 21 Jun. 2016 <https://www.scrumalliance.org/community/articles/2013/january/traditional-and-agile-methods-an-interpretation>

A really great resource that gives a brief overview of some of the Agile methodologies. Many of these we cover in the course, but some we do not (For example, Crystal). It describes Agile as an "umbrella" in which all these methodologies fall under--this is the same approach that we take in this course and specialization. Works well as an introduction or study guide!

"Agile Methodologies for Software Development - VersionOne." 2015. 21 Jun. 2016 <https://www.versionone.com/agile-101/agile-methodologies/>

This shows the results from a survey, conducted by VersionOne, that shows you the popularity of Agile, and other methodologies like Scrum, Kanban, Extreme Programming, and Lean in the software industry.

"8th Annual State of Agile Survey - VersionOne." 2015. 21 Jun. 2016 <https://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>

Extreme Programming

Test Driven Development, also known as TDD, is a foundation of Extreme Programming. If you wanted to read more about TDD, this book is an excellent guide to implementing TDD with your team. If you wanted just a quick explanation, this is not the resource for you.

"Test Driven Development: By Example: Kent Beck: 9780321146533 ..." 2014. 21 Jun. 2016 <https://www.amazon.com/Test-Driven-Development-Kent-Beck/dp/0321146530>

If you are interested in implementing XP in your team, this would be a great resource for you. This book describes what Extreme Programming is, as well as why it works and why the practices are the way that they are.

"Extreme Programming Explained: Embrace Change, 2nd Edition (The ..." 2016. 21 Jun. 2016 <https://www.amazon.com/Extreme-Programming-Explained-Embrace-Change/dp/0321278658>

A simple explanation of Extreme Programming and it's practices. Provides a good overview or study guide for XP.

"Extreme Programming (XP) FAQ - Jera Design." 2002. 21 Jun. 2016 <http://www.jera.com/techinfo/xpfaq.html>

Another simple XP guide. Explains the practices and has a few diagrams. It also gives you a bit of history about how and when traditional models transitioned into Extreme Programming and other Agile methodologies, if that is something that interests you.

"Extreme Programming." 2006. 21 Jun. 2016 <http://www.umsl.edu/~sauter/analysis/f06Papers/Hutagalung/>

This is my favorite Extreme Programming resource. It doesn't just give you the practices, it explains what they are, and how to implement them in more detail. This is especially useful if you don't understand one of the practices (i.e. System Metaphor). Note that this resource does not have the "12 rules of XP", instead they have many more that still fall within the XP methodology.

"Extreme Programming: A Gentle Introduction." 21 Jun. 2016 <http://www.extremeprogramming.org/>

Scrum

A really great video to watch if you are new to Scrum. It pretty much describes the basics of Scrum in a quick 10 minute video.

"NEW Intro to Agile Scrum in Under 10 Minutes - What is ... - YouTube." 2012. 21 Jun. 2016 <https://www.youtube.com/watch?v=XU0llRltyFM>

You can read or download (in PDF) the Scum Guide. This gives you a description of just about everything related to Scrum. This is a really great resource to print out and keep with you if you are implementing Scrum in your team.

"Scrum Guides: Home." 2013. 21 Jun. 2016 <http://www.scrumguides.org/>

Module 4 - Supplemental Resources

Agile Variations and Lean Software Development

This article explains Feature Driven Development (FDD) in good detail if you were interested in reading more about the subject.

"Feature Driven Development (FDD) and Agile Modeling." 2014. 21 Jun. 2016 <http://agilemodeling.com/essays/fdd.htm>

A really great overview of the Agile Unified Process (AUP). It offers diagrams, and explanations. It also covers if you should implement AUP.

"The Agile Unified Process (AUP) Home Page - Scott Ambler." 2005. 21 Jun. 2016 <http://www.ambysoft.com/unifiedprocess/agileUP.html>

This article goes beyond the traditional explanation of Lean. It examines some principles and values that supplement the Lean methodology and have been added since the initial inception of Lean. An interesting read if you are interested in implementing or learning more about the Lean methodology.

"Lean Software Development - MSDN - Microsoft." 2015. 21 Jun. 2016 <https://msdn.microsoft.com/en-us/library/hh533841(v=vs.120).aspx>

This examines common "waste" in Agile software development and how you can manage them in your projects. This relates to the Lean principle of "eliminating waste".

"Tips to manage the "7 wastes" of "Agile Software Development"" by Vijay Bandaru. 22 Oct. 2018 <https://www.linkedin.com/pulse/tips-manage-7-wastes-agile-software-development-vijay-bandaru/>

This is a great resource for this course! It goes through several methodolgies and gives a great overview of each. The guide covers Agile, Extreme Programming, Scrum, Lean, Feature Driven Development (FDD), and Agile Unified Process (AUP), as well as some methodologies that we don't cover in this specialization like Crystal and Dynamic Systems Development Method. On the last page of Part 2, they highlight and compare the strengths and weaknesses of each of these methodologies. This would be a great study guide for this course.

"A Practical Guide to Seven Agile Methodologies, Part 1 - DevX." 2006. 21 Jun. 2016 <http://www.devx.com/architect/Article/32761>

"A Practical Guide to Seven Agile Methodologies, Part 2 - DevX." 2006. 21 Jun. 2016 <http://www.devx.com/architect/Article/32836>

This is the ultimate guide to Lean software development. If you are interested in implementing Lean, then this is the resource you want to refer to.

"Lean Software Development: An Agile Toolkit: Mary Poppendieck ..." 2015. 21 Jun. 2016 <https://www.amazon.com/Lean-Software-Development-Agile-Toolkit/dp/0321150783>

A Wikipedia article about behaviour-driven development. This was touched upon in this module and would be a good resource if you wanted more information on the subject.

"Behavior-driven development - Wikipedia, the free encyclopedia." 2011. 21 Jun. 2016 <https://en.wikipedia.org/wiki/Behavior-driven_development>

A Wikipedia article about Dynamic Systems Development Method. This is not a necessary reading, but would be useful if you wanted to more information about the subject.

"Dynamic systems development method - Wikipedia, the free ..." 2011. 21 Jun. 2016 <https://en.wikipedia.org/wiki/Dynamic_systems_development_method>

Kanban

A good site that explains what Kanban is and how to implement it. It has good visuals.

"What is Kanban? - Kanban Blog." 2014. 7 Jul. 2016 <http://kanbanblog.com/explained/>

A brief explanation of Kanban. This includes it's history, how it's used today, and the principles.

"What is Kanban? - LeanKit." 2016. 7 Jul. 2016 <https://leankit.com/learn/kanban/what-is-kanban/>

A Wikipedia article that explains Scrumban, which is the common term for when Kanban is combined with the Scrum methodology.

"Scrumban - Wikipedia, the free encyclopedia." 2015. 21 Jun. 2016
<https://en.wikipedia.org/wiki/Scrumban>