Uploaded lecture on projects day will have a 10 mark guaranteed Q.

Project Info:

Google link uploaded to eve: Everyone must fill in.

Dev guide: Like javadocs, explain all functions… just must be readable  
User guide: All dependencies and software requirements.  
Installation guide  
---need to all be done by projects day.

Marketing website: Will receive an email guide… Due Monday

Short desc of team + project, 100 char.

**All work from this day forward 02/10/2023 is guaranteed in exam.  
This chapter has a 10 mark question at least.**

Change is inevitable. A project scope will always change.

Focus changes as project is developed.

You need good stats. Your stats showcase the understanding of domain and project, whether you’ve produced the correct project, and the relevancy of your system.

Changes need to be calculated, analysed. How big does this change cause to the system. Do the risks of the change have high enough benefits? Drastic changes are not made near the end of the lifecycle, or near releases (as you do not want to break your system).

**Software configuration Management(SCM)**

**Applied throughout software development process.**

**They are developed to:**

1.Identify Change

2.Controls Change

3.Ensure that the change is properly implemented

4.Reports changes to others who may have an interest in the change: Such as project managers, developers ect.

**The output of the software process can be seperated in 3 broad categories**

1.Computer programs (both source level and executable forms)

2.Work projects that describe the computer programs (targeted at various stakeholders)

3.Data pr content (contained within the program or external to it)

Change may occur at anytime, and for any reason.

**Need to know this quate: NBNB**  
“The first Law of Software Engineering”  
“No matter where you are in the system life cycle, the system will change. And the desire to change it will persist…”

**There are 4 fundamental things that cause change**: (**Fundamental NBNBNB [know the name]**, not the only causes)

1.New business or market conditions dictate changes in project requirements or business rules.   
-The rate of change effects development. Eg Agile for high changes.

2.New stakeholders needs demand notification of data produced by information systems, functionality delivered by products, or services delivered by a computer based system.  
-When a company stakeholder change, they might want to put their own stamp on the project, their own personal twist.

3.Reorganisation or business growth/downsizing causes changes in projects priorities or software engineering team structure.

4.Budgetary or scheduling constraints cause a redefinition of the system or product.  
-Most important, as no resources or if system is not relevant at end, there’s no point in project.  
-Cutting down schedule: do you hire more or cut down features.

**Exam question when you have a SQA item constantly changing, how does the SQA  
ON software, development, requirements**

**There are 4 important elements that should exist when a configuration system is developed:**

**Component elements:** A set of tools coupled within a file management system that enables access to and management of each software configuration. **-**Managing individual components

**Process elements**: A collection of procedures and tasks that define an effective approach to change management for all constituencies involves…   
-this is where SQA implemented

**Construction elements:   
-**Once you have configuration, constructs everything to execute code

**Human element:** A set of tools and process feature used by the software team to implement effective SCM.   
-This is where the make or break of the project.

**The SCM repository**

While implementing changes, cataloguing and analysing everything, you need somewhere to store everything.

A set of mechanisms and data structures that allow teams to manage changes in an effective manner.

Provides the obvious modern DB system, Data integrity and providing and sharing of DB software tools. (GIthub)

**A robust repository provides 2 types of broad services:**

1.The same types of services that might be expected from any sophisticated management system

2.Services that are specific to the software engineering system

**A repository that serves a software engineering team should also:**

1.Integrate with or directly support process management functions.

2.Support specific rules that govern the SCM function and the data maintained within the repository.

3.Provide and interface to other software engineering tools. (eg git in VS)

4.Accommodate storage of sophisticated data objects.

**To support SCM, the repository must have a tool set for:**

**Versioning**: As a project progresses, may versions of individual work products will be created.  
-Rollbacks to previous working versions.

**Dependency tracking and change management:** The repository manages a wide variety of relationships among the data elements stored in it (Branching).

**Requirements tracing:** The special function depends on link management and provides the ability to track all the design and construction components and deliverables that result from a specific requirements specification.

-**Configuration Management:** A configuration management facility keeps track of a series of configurations representing specific project milestones or production releases.

**Audit trails:** An audit trails establishes additional information about when, why and by whom changes are made.  
-Commit messages.

**Software configuration process defines a series of tasks that have 4 primary objectives:**

1. To identify all items that collectively define the software configurations
2. To manage changes to manage change between one or more of these items (For unit testing)
3. To facilitate the construction of different versions of an application.
4. To ensure that software quality…….

**Structure of SCM Repository**

Start with:  
**SCI:** The object/function going in system. (**new branch added**)

**Identification:** Must first identify the SCI, what’s its purpose, how does it get in system (identify changes on branch that need to be made)

**Change control:** How does change occur **(Make change)**

**Version Control:** The version of a function is important. Can go back and test previous versions of functions and compare them. (**Berge branch to main with version**)

**Configuration auditing:** Maintain integrity of the software being built. So that what is being build, is being built correctly. (**Commits for branch**)

**Reporting:** Now that everything’s done, you can report to stakeholders that items finished and ready for use. Reporting what has changes, new features, optimisation to Stakeholders to keep them up to date.