watched videos and these are the key insights:-

1. Test Driven Development

->Some choices may make things better for clients and some may not. We have to decide whether
->Spend right amount of time asking questions and understanding and using TDD as a vehicle to
->Choose tools wisely.

V-Model

->Its advantage is for every major activity testing is done.
->Its disadvantage is testing is done after development has finished.
->The longer one waits for testing, more work is done, if the test fails work must be discarded.

Royee's model

->After each phase or step testing is done.
->If a test is failed we can go one step backward.
->Just because the test passes doesn't mean that is good. That test may have flaws.

Key Questions:

->Should I try to gather all requirements?
->How much design/documentation is enough?
->When I do stop producing detailed design and start coding?
->If I do too much, I have to discard some work.
->If I do too little I must refactor too ofte-
->Powerful Tools
->Pair programming.
->Black box testing- writing test before writing code.
->White box testing- Testing after coding
->Create throw away prototypes to verify the requirements.
->Provide to demos to clients regularly.
->Creating the tests first forces us to deeply understand the work before we actually do it.
->Getting others to help us create the tests and verify the results improves the chances for quality a

2.Introduction to requirements

->Requirements engineering is about knowing what client needs and how to satisfy them.
->Elicitation- gathering information and prioritizing needs.
->We have to ensure we satisfy all of the stakeholders especially who have power.
->Understanding recording and monitoring the requirements is crucial.
->Remain neutral and should not bias the input.
->Establish the business goal first.
->Work down from these business goals to technical goals.
->Should stay in the problem space - gather as much information as you can.
->Information can be gathered by surveys, interviews, questionaries', group methods.
->By the end of elicitation- deep understanding what the problem is, what needs to be changed or
->Modelling is a tool to understand the things we believe are important and utilize them.

->it is partitioning of some real world systems into a collection of interacting components.

Various business models
1. Enterprise model
2. Data model
3. Behaviour model
Requirement should also specify how to verify that it has been properly addressed once the system is bu

3.Risk management:
->Risk management is a critical part of any complex software project.
->number of problems increase with the size of the project.
Understanding the problems:
->Common cause is based on active elements in the system and can be avoided.
->Special cause is new,emergent,difficult to predict and diificult to avoid.
Constraint:
A constraint is something we must do as there is 100% probability of loss if we fail to act.
Risk:
A risk is potential for a loss.
Risk categories:
->project risks- unable to build
->technical risks - unable to make it work
->business risks - unable to make money at it.
Project Management:
->Careful attention to risk management is required.
->Risk management is a structured process of recognizing and addressing problems that are like

4.created product backlog for our project

5.State diagrams:
->Describe system behaviour
->determines what is possible, what is not possible and suggests things which are useful to us.
->Arrows indicate the direction of transition from one state to another

Last Saved--Dec 20, 2014 12:17:40
--Last Saved--Dec 29, 2014 10:08:23
--Last Saved--Dec 29, 2014 15:17:52
--Last Saved--Dec 29, 2014 15:17:59
--Last Saved--Dec 29, 2014 15:18:04
--Last Saved--Dec 29, 2014 15:33:39
--Last Saved--Dec 29, 2014 15:44:45
--Last Saved--Dec 29, 2014 15:52:34

Deliverables

Delivearables				
Test Driven Development	know about Test Driven development	what is test driven development	21:57 mins	27 mins
Introduction to Requirements	what are requirements	requirements for the project	32 mins	40 mins
Create product backlog	creating product backlog for he project	created product backlog	1 day	2 days

Lessons Learned

Reflections

Plan for next week

Plans	
Hint	Display hints when the screen is idle and selected answer is wrong
Lives	One life should be deducted for choosing one wrong answer
Creating objects	To make the code object oriented