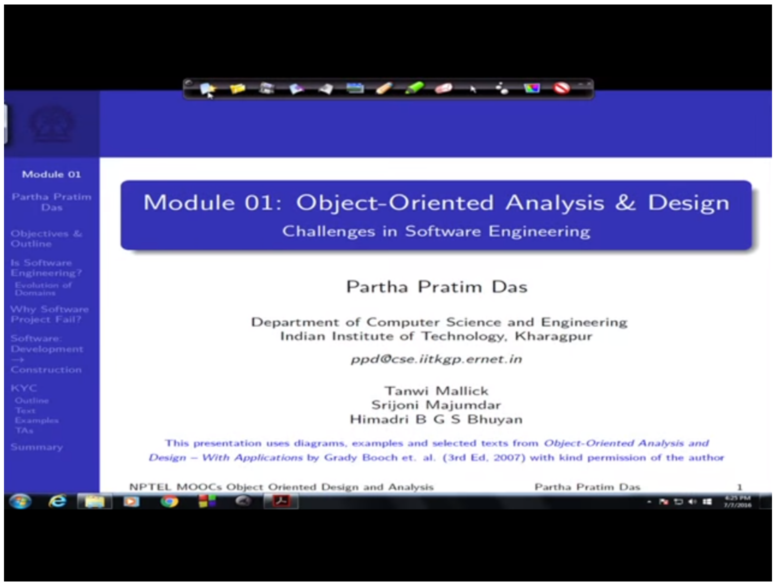
Object-Oriented Analysis and Design

Prof. Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 01
Challenges in Software Engineering

Welcome to the moocs course on object-oriented analysis and design. This is module 1.

(Refer Slide Time: 0:46)



Module 01
Partha Pratim Das
Objectives & Outline
Is Software Engineering?
Evolution of Domains
Why Software Project Fail?
Software: Development → Construction
KYC
Outline
Text
Examples
TAs
Summary

Module 01: Object-Oriented Analysis & Design
Challenges in Software Engineering

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ernet.in

Tanwi Mallick
Srijoni Majumdar
Himadri B G S Bhuyan

This presentation uses diagrams, examples and selected texts from *Object-Oriented Analysis and Design – With Applications* by Grady Booch et. al. (3rd Ed, 2007) with kind permission of the author

NPTEL MOOCs Object Oriented Design and Analysis   Partha Pratim Das   1

Here, we will start by setting the background and creating the motivation for, why is it required to study understand object-orientation? How does it plea with analysis of real life problems and how it can be effectively used in design of complex software. Certainly you all are familiar with object-oriented programming; at least I assume that you know glimpses of that. You aware that C++ or Java are some of the most popular object-oriented programming languages.

Now the fact of the matter is that while you design, implement and maintain a complex software, certainly using a object-oriented programming language turns out to be of great help but merely the use of a object-oriented programming language, that a programming language which supports definition, creation, management, destruction of objects, merely the use of that language as a programming vehicle does not solve all the issues of software system.

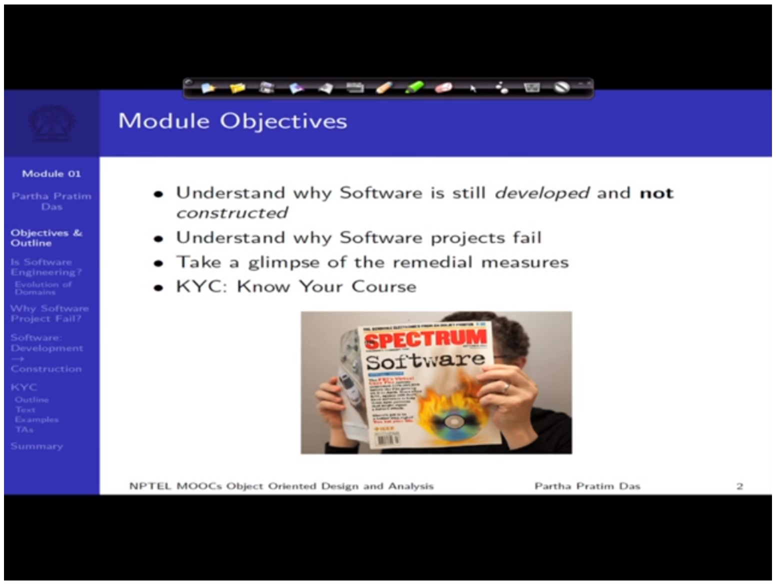
And in reality, much of the problems that software systems encounter or the developers and as well as a users of the software systems encounter and not just limited to the programming code. Several actually significant part of the problems start at the specification of the problem itself, where the user talks to the developer and wants to express that this is what I want as a software. Now in some cases, users will directly do that if you are have an enterprise and you are trying to create,

Get a system created by a software vendor, say to manage your employees, to manage the leaves that your employees take, you will directly talk to the vendor and get a system created for your organizational use or it could be that it is a generic software product where several independent unknown users have talked to several parts of the developers to express their requirements. And all those requirements had been canned into a complete set, a consistent set to develop the product.

An example could be Microsoft word, excel, this office (())(03:32) and mail clients, browsers, they have not been developed for any specific needs of any specific organization or individual. They have been created for generic use by a large number of individuals, groups and organization and therefore we call them the products, irrespective of which kind of software system. We talk about the specification, design, analysis and finally implementation, maintenance, preservation are all critical issues.

That needs to be effectively handled, particularly for a industrial strength large scale software. So you would be familiar that this whole exercise is a prescient in the name of software engineering like civil engineers, construct, mechanical engineers construct automobiles, machines. Electrical engineers construct power plants, air conditioners, fans, software engineers are supposed to build software and that discipline is known as software engineering. So to understand the requirements.

Or rather more the motivation of why we need to engage the object-oriented analysis and design, we will start by taking a brief look into, what are the challenges that software engineering or software engineers face, that will be the main state of our current module.
(Refer Slide Time: 05:09)


Module Objectives

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?
Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC
Outline
Text
Examples
TAs

Summary

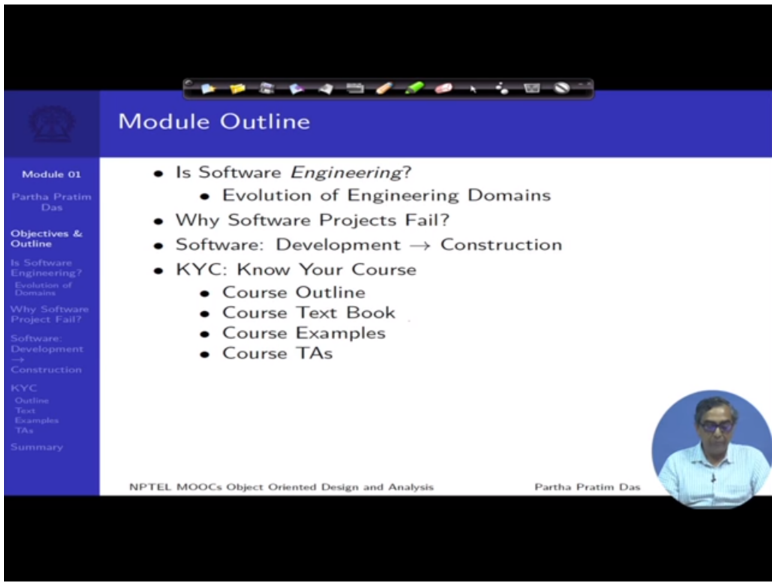
- Understand why Software is still *developed* and **not** *constructed*
- Understand why Software projects fail
- Take a glimpse of the remedial measures
- KYC: Know Your Course

NPTEL MOOCs Object Oriented Design and Analysis — Partha Pratim Das — 2

So to specify, the specific objective would be to understand why software is still developed and not constructed. Please note these two words development and construction. If you tell into any field of engineering other than software, you will typically find that the word used at the, to define the activity to define the product is called constructions. Buildings are constructing, power plants are constructing but software is typically developed, not constructed.

So we will take a little bit of round in terms of why this difference in semantics. We will try to understand why software projects fail and take a glimpse into some of the remedial measures and finally, we will end the module with the KYC, that is know your course. That is what we are going to cover in this whole.

(Refer Slide Time: 06:10)


Module Outline

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?
Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC
Outline
Text
Examples
TAs

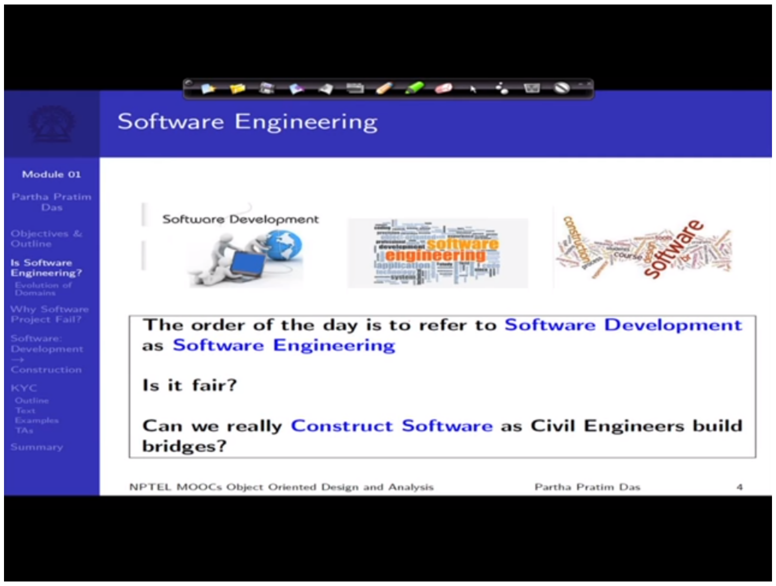
Summary

- Is Software *Engineering*?
  - Evolution of Engineering Domains
- Why Software Projects Fail?
- Software: Development → Construction
- KYC: Know Your Course
  - Course Outline
  - Course Text Book
  - Course Examples
  - Course TAs

NPTEL MOOCs Object Oriented Design and Analysis — Partha Pratim Das

The outline would be is software. What is software engineering? Or rather is software really engineering and why projects fail and so on. And the convention we will typically follow in all the modules is during the presentation, you will be able to see the outline of the module on the left hand side of every slide and the current subsection in the outline that is being discussed in a given slide will be highlighted on the left.

(Refer Slide Time: 06:43)



Software Engineering

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

Summary

The order of the day is to refer to Software Development as Software Engineering

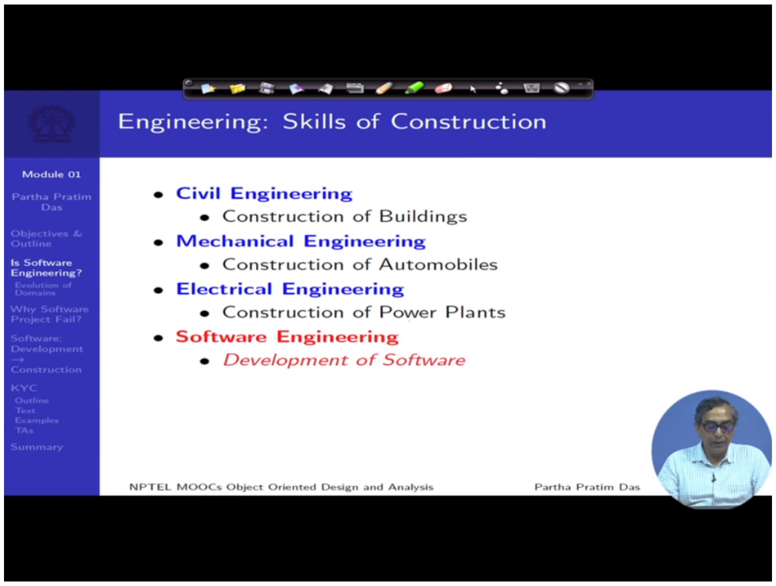
Is it fair?

Can we really Construct Software as Civil Engineers build bridges?

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 4

So this is the outline that you will have. So now let us look at software engineering, so the order of the day is to refer to the activity of the software development as software engineering. I start by raising a question that is it fair to really pronounce proclaim that we are doing software engineering? If you are doing software engineering then we must be able to construct software as civil engineers build bridges.

(Refer Slide Time: 07:12)



Engineering: Skills of Construction

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

Summary

- Civil Engineering
  - Construction of Buildings
- Mechanical Engineering
  - Construction of Automobiles
- Electrical Engineering
  - Construction of Power Plants
- Software Engineering
  - *Development of Software*

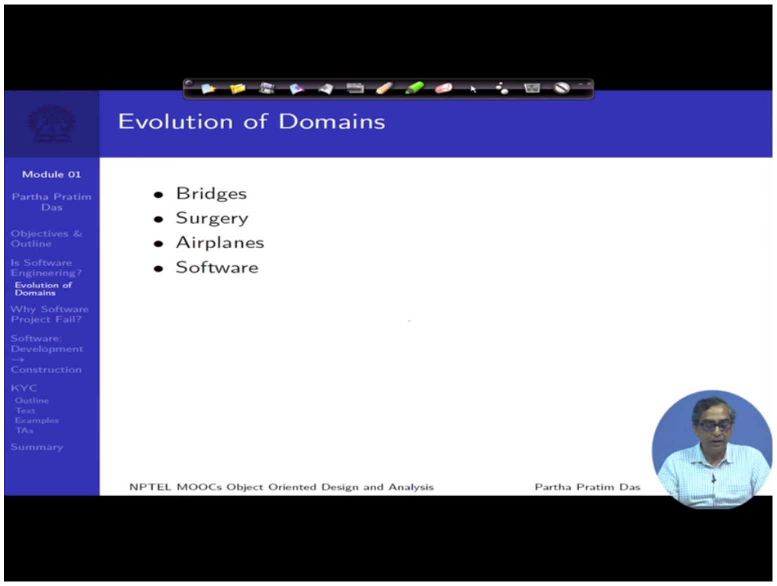
NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

So what is engineering? Engineering in very simple terms can be look at from different angles but the core concept is, it is a scale of construction. Engineering also deals with the ability to manage complexity, what is the difference between science, theory and engineering? The science tells us the natural laws, principles, mathematics that govern, how systems will behave. We turn them into engineering when we add utility to what we do one or more people one or more organizations get benefited.

And the second very important factor is we create something which did not exist and in the whole process we really need to learn engineering because there is a complexity to handle if things are small, simple then you do not need to be an engineer. You can just want to analyze as you throw a ball and want to hit a target, what is the force with which you will throw the ball and hit the target. You can do that from your study of physics by using Newton's law of motion.

But when you want to do that with by firing a missile which has to go over 5000 kilometers and still hit a building at right point, you need to do engineering. Both are actually hitting a target over a distance but what is different in case of missile is, the complexity is very different. So when we talk about engineering we basically talk about how to manage complexity and that the core skill of an engineer is to learn how to what is the complexity and how to handle it.

So just to take a glimpse we will take a quick tour this is more of a story telling a quick tour into some of the common fields of engineering and see how they have evolved.
(Refer Slide Time: 09:22)

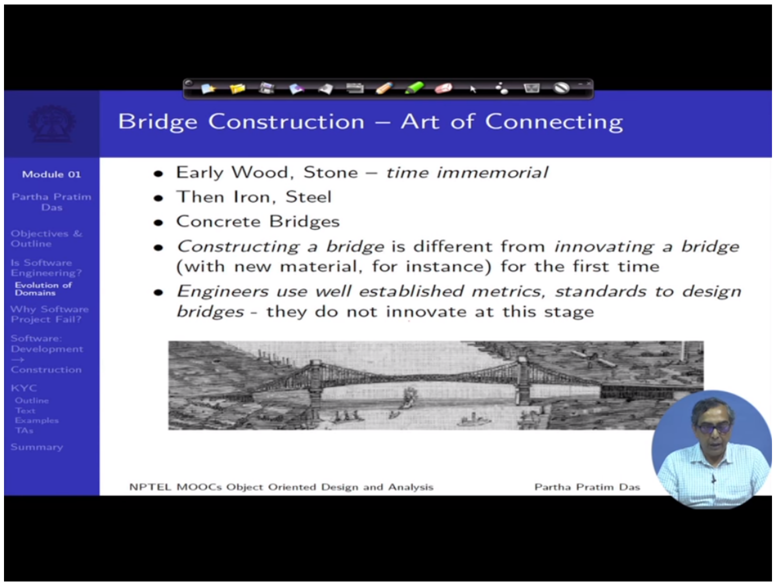


Evolution of Domains

- Bridges
- Surgery
- Airplanes
- Software

So we will start with this 4 different domains the bridges the surgery aero planes and certainly software.

(Refer Slide Time: 09:31)



Bridge Construction – Art of Connecting

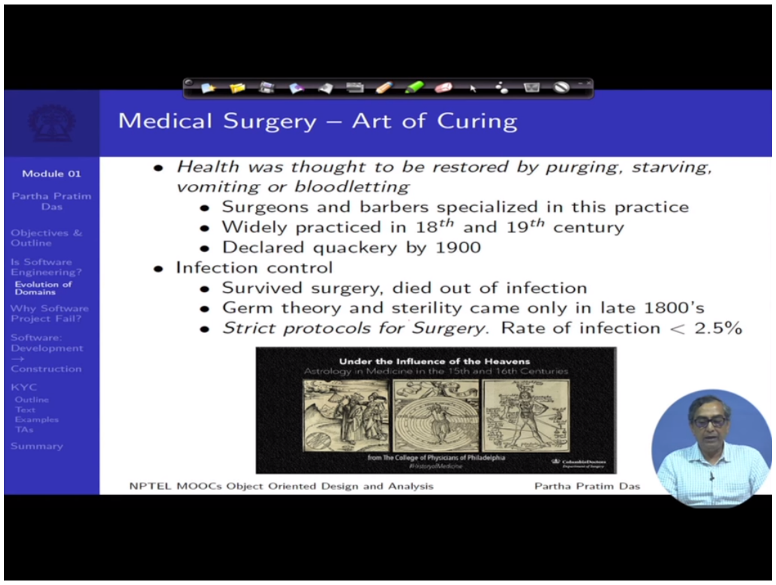
- Early Wood, Stone – *time immemorial*
- Then Iron, Steel
- Concrete Bridges
- *Constructing a bridge* is different from *innovating a bridge* (with new material, for instance) for the first time
- *Engineers use well established metrics, standards to design bridges* - they do not innovate at this stage

So if you look into bridges, this is one of the earliest constructions that the human civilization had to do from time immemorial we have been constructing bridges, first with wood, stone, then iron, steel, glass, concrete bridges and so on and by now if you are given to construct a bridge then we do not accept this outcome that the bridge may fall that the construction may not succeed. If I construct a flyover I would not expect the flyover to fall over. Let us not talk about incidents in Delhi and Kolkatta that we have seen in last couple of hours.

That is faults in engineering but the civil engineering knowledge guarantees that the bridges will not fall. So constructing a bridge is different from innovating a bridge. We will say we are innovating on a bridge if we are experimenting a new material, we have seen bridges with iron, steel, concrete but then if you look at the Google for Grand Canyon, you will find bridges which are completely in glass. So when civil engineers sit down and find out how a bridge can be constructed completely with glass.

And experimenting or the innovating a bridge but when they are actual engineers and constructing a bridge then they are working with established metrics standards proof and processes so that is there is no innovation particularly and the process must succeed.

(Refer Slide Time: 11:17)



Medical Surgery – Art of Curing

Module 01
Partha Pratim Das
Objectives & Outline
Is Software Engineering?
Evolution of Domains
Why Software Project Fail?
Software: Development → Construction
KYC
Outline
Text
Examples
TAs
Summary

- *Health was thought to be restored by purging, starving, vomiting or bloodletting*
  - Surgeons and barbers specialized in this practice
  - Widely practiced in 18th and 19th century
  - Declared quackery by 1900
- Infection control
  - Survived surgery, died out of infection
  - Germ theory and sterility came only in late 1800's
  - *Strict protocols for Surgery*. Rate of infection $<$ 2.5%

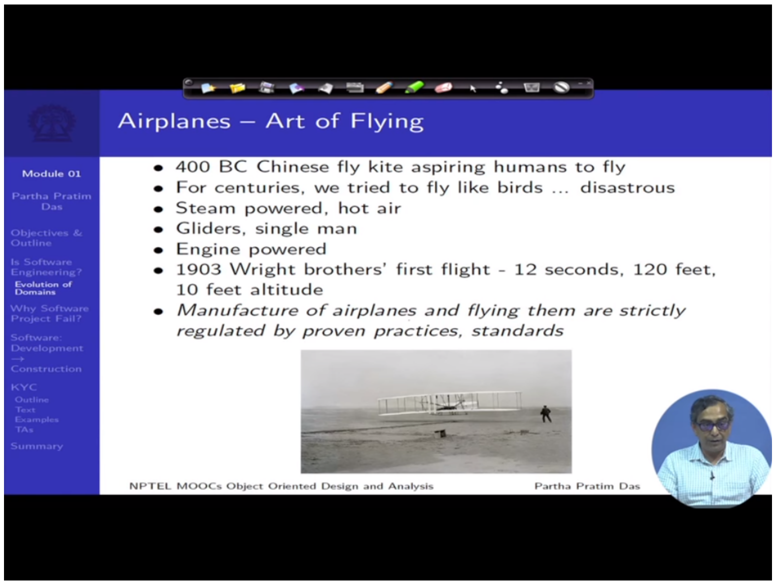
Under the Influence of the Heavens
Astrology in Medicine in the 15th and 16th Centuries
from The College of Physicians of Philadelphia
#HistoryofMedicine

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

The same story will reveal itself in other fields, just consider medical surgery. Health naturally has been a issue of concern again from time immemorial and as is documented that surgery and the act of barbers were considered equivalent. Prior to 18th century you had to cut the human body and do something it was declared quackery by 1900 but people slowly, the doctors or surgeons started perfecting on their surgery but most patients did not survive because of the injury infection.

So we learnt how to manage infection in the late 1800s, now if you look into the way the surgery is done every surgery follows the strict protocol. The protocol is nationally or internationally be established proven and the doctor and the team has to specifically follow the protocols to ensure that the surgeries succeeds, the infection is not propagated to the patient and so on and that is the reason that surgery today has become so success. So that is also a kind of engineering that we see.

(Refer Slide Time: 12:41)



Airplanes – Art of Flying

- 400 BC Chinese fly kite aspiring humans to fly
- For centuries, we tried to fly like birds ... disastrous
- Steam powered, hot air
- Gliders, single man
- Engine powered
- 1903 Wright brothers' first flight - 12 seconds, 120 feet, 10 feet altitude
- *Manufacture of airplanes and flying them are strictly regulated by proven practices, standards*

Module 01
Partha Pratim Das
Objectives & Outline
Is Software Engineering?
Evolution of Domains
Why Software Project Fail?
Software: Development → Construction
KYC
Outline
Text
Examples
TAs
Summary

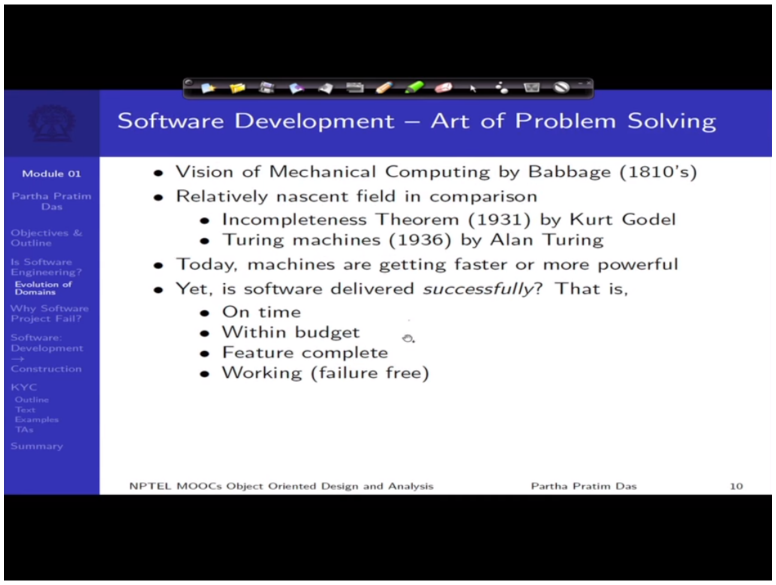
NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

And you can see that the success rate is quite high and without that of course we would be very scared to go to the surgeons. The third domain is aero planes many of us is certainly love flying and again the attempt started at around 400 BC with Chinese trying to flying kite and aspiring that humans could also fly. For centuries people has tried to fly as birds with disastrous effects many people have lost their lives.

There have been different means of flying, steam powered, hot air gliding for single man, engine powered and so on. Till about little over a 100 years back the wright brothers finally made the today's prototype of aero planes and they could fly again if you look at how the aero planes are manufactured how they are flown is strictly regulated governed by proven practices, standards and if those standards are not adhered to then the aero plane will not or the aircraft will not be allowed to fly the pilot will not be able to take the seat and so on.

And that gives us the comfort the guarantee that if I fly in an aircraft then I will reach the destination safely. So in all of these fields the underlined factor is the the confidence or the surety or the guarantee of the success is very very very high and often when things go wrong when things fail they are they fail either because some human being has made a mistake or many several human beings have made some mistake or there is some intentional wrong doing.

(Refer Slide Time: 14:27)


Software Development – Art of Problem Solving

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

Summary

- Vision of Mechanical Computing by Babbage (1810's)
- Relatively nascent field in comparison
  - Incompleteness Theorem (1931) by Kurt Godel
  - Turing machines (1936) by Alan Turing
- Today, machines are getting faster or more powerful
- Yet, is software delivered *successfully*? That is,
  - On time
  - Within budget
  - Feature complete
  - Working (failure free)

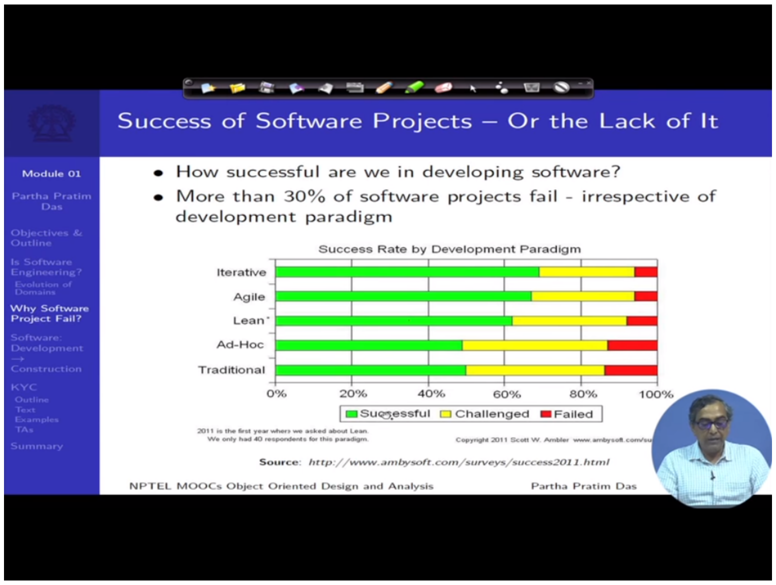
NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 10

In contrast let us look at software development I love to call it a part of problem solving the vision for computing mechanical, computing, automated computing started in early 1800 with Charles Babbage and Adalh and so on but incidentally there was not much development for over a hundred years afterwards. The first significant step towards software algorithms and software systems happened in the 1930s with the 2 seminal works, 1 by Kurt Godel's incompleteness theorem and alan turing in turing machines many of you have been heard of this.

So if you look at then software is not of 100 years old so its its compared to other fields of engineering its very very nascent, very very new and therefore it has not got a time really prove its processes, setup, its standards and make sure that the success will always be at the end of the route, of course we see a lot of developments happening very fast developments machines are expanding, they are getting faster more powerful, lot of machines yet the software delivered successfully we all understand that it is not.

And let me for the sake of being formal in this course define what I mean by the successful delivery of the software it means primarily 4 factors that the project is completed one time. As it was planned there is no extension of the project, it is completed within budget that is there is overshoot of funds, all features that we have promised are completed. It is not that I will do it on time within budget and implement only 50% of the functionality that I wanted and finally it must be working software that is it should be failure free fault free.

(Refer Slide Time: 16:24)



Success of Software Projects – Or the Lack of It

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

Summary

- How successful are we in developing software?
- More than 30% of software projects fail - irrespective of development paradigm

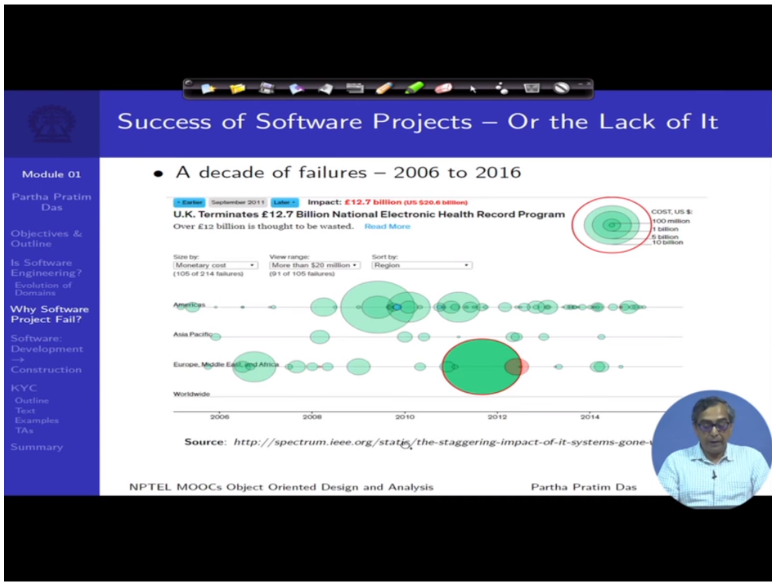
Success Rate by Development Paradigm
Iterative
Agile
Lean*
Ad-Hoc
Traditional
0%
20%
40%
60%
80%
100%
🟩 Successful 🟨 Challenged 🟥 Failed
2011 is the first year where we asked about Lean. We only had 40 respondents for this paradigm.
Copyright 2011 Scott W. Ambler www.ambysoft.com/su

Source: *http://www.ambysoft.com/surveys/success2011.html*

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

Now in this background if you look at some of the recent studies these I have picked up from the moses governance, these are 5 years to absolutely recent studies which show that more than 30% of the software projects fail and these this statistics also taken for really large projects if you take into medium size projects, the success rate will be low so that is quite alarming.

(Refer Slide Time: 16:51)



Success of Software Projects – Or the Lack of It

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

Summary

- A decade of failures – 2006 to 2016

Impact: £12.7 billion (US $20.6 billion)

U.K. Terminates £12.7 Billion National Electronic Health Record Program

Over £12 billion is thought to be wasted. Read More

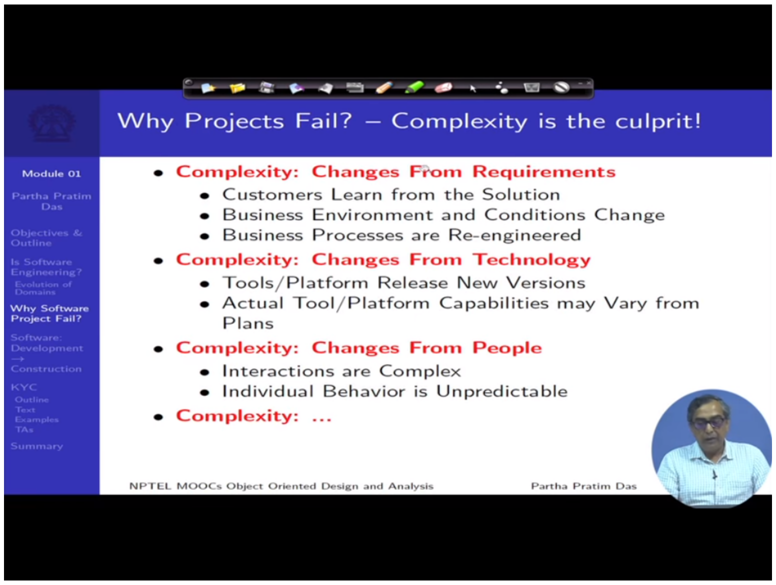
Source: *http://spectrum.ieee.org/static/the-staggering-impact-of-it-systems-gone-*

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

And this is another data which is very very recent, couple of days back which show that over the last decade 2006 to 2016 again, what are the different number of large projects that have failed. So here this bubbles this circles show the kind of budget that that particular software had. So the selected one that you can see here this one is actually 12.7 billion pound project which had been dominated by the UK government little around the middle of 2011.

And this show that there is this I mean irrespective of which region of the world you are talking of there are series of large to very large to huge software projects, which are failing causing several damages to property loss of money and effort and resources so that that is the basic alarm point.

(Refer Slide Time: 17:58)



Why Projects Fail? – Complexity is the culprit!

- **Complexity: Changes From Requirements**
  - Customers Learn from the Solution
  - Business Environment and Conditions Change
  - Business Processes are Re-engineered
- **Complexity: Changes From Technology**
  - Tools/Platform Release New Versions
  - Actual Tool/Platform Capabilities may Vary from Plans
- **Complexity: Changes From People**
  - Interactions are Complex
  - Individual Behavior is Unpredictable
- **Complexity: ...**

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

So we has tried to we will talk about this lot more but here in the in this introductory part I will just quickly try to highlight why software projects fail that can be one line answer. The complexity is the culprit this complexity of several dimensions in software development. So some of them you can you can very quickly understand is the change of requirement when you start developing the software you are told something.

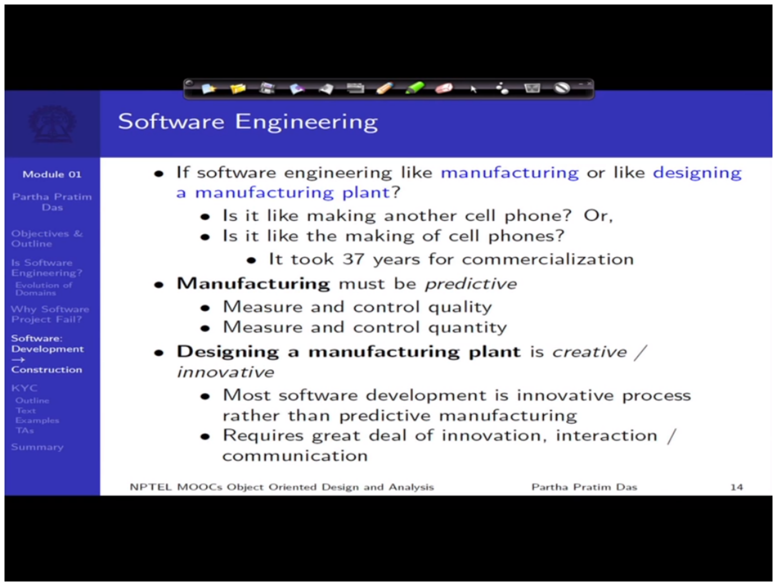
And then as you go through the development the customer would repeatedly come back to you and say that I want something different. Now is it that the customer is not a smart person is it that the customer is not being truthful to you? No please do not misunderstand the customer. The customer is changing the requirement because the customer originally did not know what she wanted she had only had a vague idea. And as you start building the software you show the prototype you show the design you show the first alpha version.

The customer gets more and more ideas, the business environment change, the condition change, processes are really changing. So one complexity big part of complexity come from the change of requirements, the different complexity comes from change from technology, software works on system, works on hardware, works on network components, works on different operating systems, third party systems and so on.

Over the period of your development if it is 1 year, 2 years the target systems that started working with that system itself evolves and the new system is around this. When you started it was windows 8.1, by the time you are in the middle of the project its windows 10, things have changed. The things may change in terms of hardware so being able to adapt to those adds a new complexity to those projects and there is a several comp severe complexity from people.

Unlike most other fields of engineering where the process of construction itself can is significantly automated. Here in software the process of construction has automation aid but its finally people who develop software, who write the code, so introduction of people as you will all understand are complex. 2 people have certain kind of interaction, when you put 4 people you get 4, say 2, 4 choose 2 kind of interactions and you have more and more people more and more groups more and more organizations more and more distributed development teams across geographies and so on introduction gets really complex.

And people are some 1 factor in the whole software industry which have an unpredictable behavior, you may everybody have certain uncertainty in terms of how you will behave today and how you will behave tomorrow. We say am in a not in a right mood, I may not be in the 100% fitness and so and so. Complexities in software development come from all of this different dimensions this is not an exhaustive list. We will talk little bit more about of the software complexity in the next module but this is just to give you glimpse in terms of why software projects fail and we will have to live with that.
(Refer Slide Time: 21:21)



Software Engineering

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

Summary

- If software engineering like manufacturing or like designing a manufacturing plant?
  - Is it like making another cell phone? Or,
  - Is it like the making of cell phones?
    - It took 37 years for commercialization
- **Manufacturing** must be *predictive*
  - Measure and control quality
  - Measure and control quantity
- **Designing a manufacturing plant** is *creative / innovative*
  - Most software development is innovative process rather than predictive manufacturing
  - Requires great deal of innovation, interaction / communication

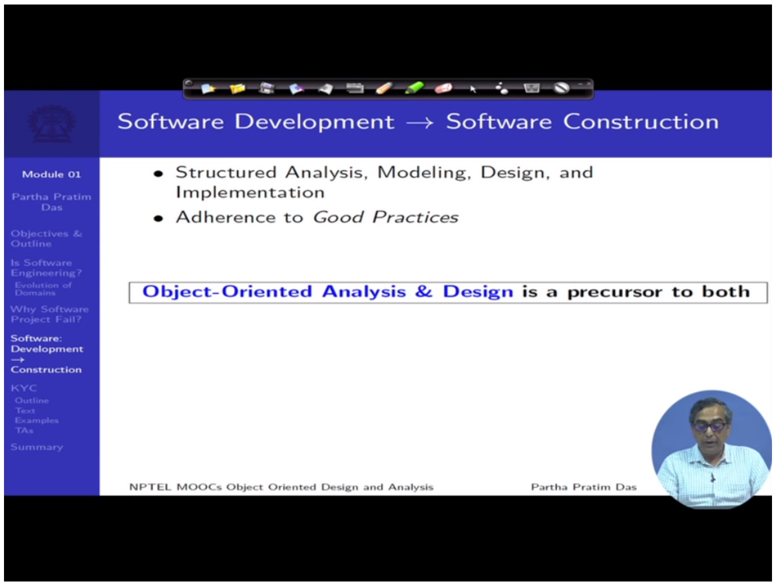
NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 14

So the question the more question is, is software engineering is like manufacturing where your have proven process and you just make a cell phone or it is like designing a manufacturing plant or like making a cell phone. So these are 2 different aspects 1 is when somebody started deciding that or dreaming that there should be cell phones, cellular phones which should be usable anywhere. To the point when today you start making Samsung galaxy phone, iphones and so on.

The scenarios these 2 scenarios are different and for your statistics it took over 37 years to make commercial cell phones, so now the manufacturing process is predictive. It must be predictive, we must be able to measure and control quality as well as quantity but designing the manufacturing is a creative innovative process, interestingly most software development is continuously innovative process rather than predictive manufacturing. So that is what makes it difficult to view software building software development as a construction process because it does not have the reliability productivity of the manufacturing of construction.

It requires great deal of innovation interaction communication and all those factors impact the success of the projects. So we have a long way to go from the status of software development to the status of software construction.

(Refer Slide Time: 22:52)



Software Development → Software Construction

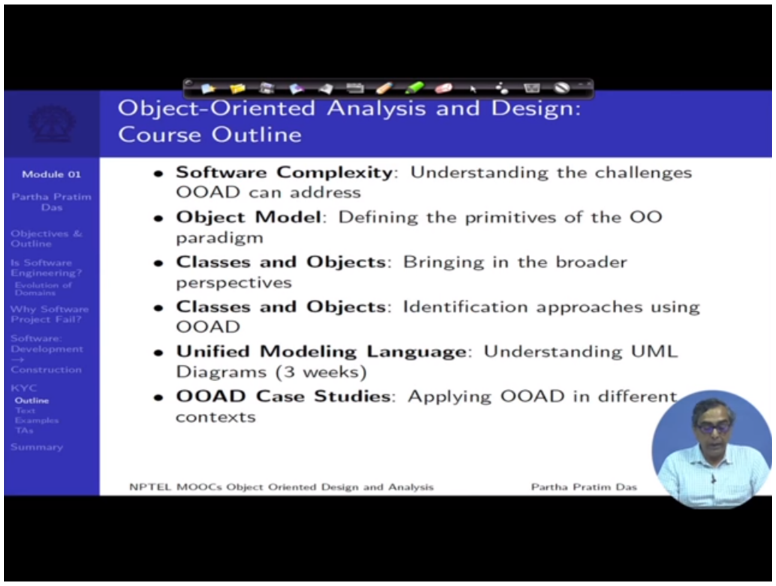
- Structured Analysis, Modeling, Design, and Implementation
- Adherence to *Good Practices*

Object-Oriented Analysis & Design is a precursor to both

We look forward to a day when like civil engineers, electrical engineers will be able to say that we have constructed a software and if you have constructed a software, it will work the process will predictable quality, will be predictable quantum, will be predictable cost, will be predictable and we will adhered to all of that. The present state of the art says that this transition of art is structured analysis modeling design and implementation.

We will slowly understand through this course what these terms mean in depth and it also needs that we are we can adhere to good practices, if you look into any of the streams of engineering there is a large volume of handbooks and standard data and practices which the engineers must follow that. That is the code of conduct I mean, u just cannot evolve or just deviate from that software engineers are still not up to it. So we are still struggling even to follow the coding guidelines in programs.

So the fact of the matter is that object-oriented analysis and design is a precursor to achieving both these major tasks which will take us forward somewhat closer to software construction than the state of development that we are in and that is the main motivation of attending discussing this course.
(Refer Slide Time: 24:25)



Object-Oriented Analysis and Design: Course Outline

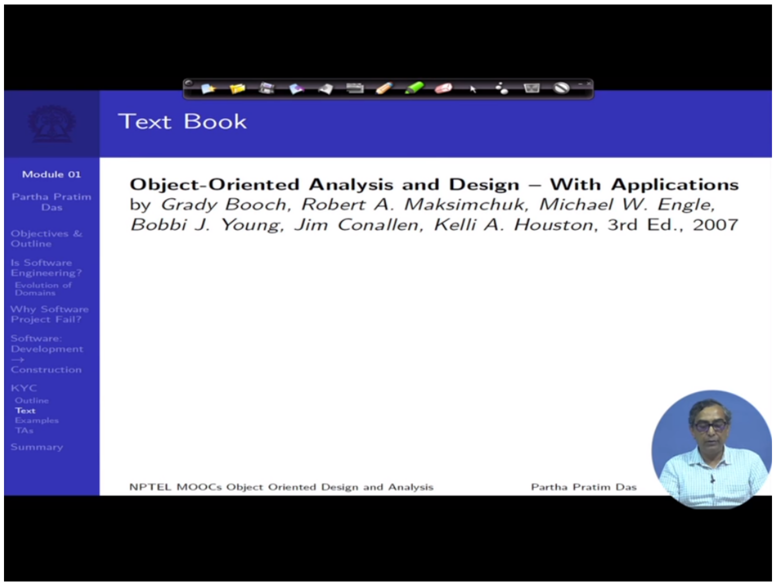
- **Software Complexity**: Understanding the challenges OOAD can address
- **Object Model**: Defining the primitives of the OO paradigm
- **Classes and Objects**: Bringing in the broader perspectives
- **Classes and Objects**: Identification approaches using OOAD
- **Unified Modeling Language**: Understanding UML Diagrams (3 weeks)
- **OOAD Case Studies**: Applying OOAD in different contexts

So before I close this module I quickly take a look into knowing your course the course outline. We will talk about software complexity, understanding the challenges that OOAD can address, will discuss about object models how to define the primitives of the object-oriented paradigm, will talk about classes and objects in terms of bringing in the broader perspective of modeling and we will talk about how with all these definitions and background of theory

We will actually do hands on exercise and give you concrete principles in terms of how you can really identify what are the approaches to identify objects and how to analyze them how to design based on them. And in this whole process, we will start using it language a graphical language which is kind of a de facto standard to express object-oriented systems today known as a unified modeling language and for 3 weeks we will learn this language we will take several examples in that language.

And will show you exactly how to use this, how to throw up in the lingua franca of object-oriented community, finally before we end we try to discuss and make you work through some of the case studies where OOAD will be applied to different contexts.

(Refer Slide Time: 25:58)



Text Book

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

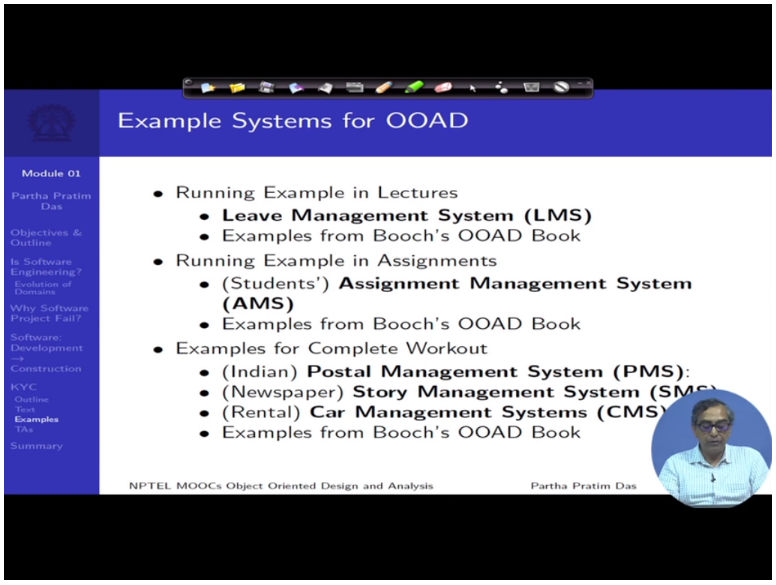
Summary

Object-Oriented Analysis and Design – With Applications
by *Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen, Kelli A. Houston*, 3rd Ed., 2007

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

This is the text books just to there are several books in object-oriented analysis design uml and so on just to keep you focused and certainly I would like to recommend you to follow this book as a text book. This is certainly in whatever I have seen this is the best book on this topic written by several stalwarts sphere added by Grady Booch during a verdict and he is with IBM now and please follow this book. This has most of the materials that we will discuss is will be borrowed from this book and also the whole discussion of UML has good summary in this book and if you follow this am sure you will find this course quite easy to go.

(Refer Slide Time: 26:52)



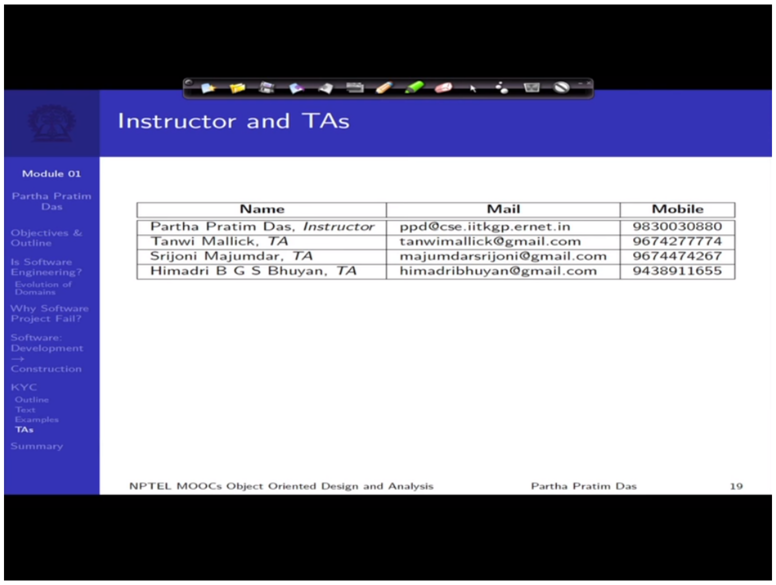
Example Systems for OOAD

- Running Example in Lectures
  - **Leave Management System (LMS)**
  - Examples from Booch's OOAD Book
- Running Example in Assignments
  - (Students') **Assignment Management System (AMS)**
  - Examples from Booch's OOAD Book
- Examples for Complete Workout
  - (Indian) **Postal Management System (PMS)**:
  - (Newspaper) **Story Management System (SMS)**
  - (Rental) **Car Management Systems (CMS)**
  - Examples from Booch's OOAD Book

These are some of the examples that we will try to follow all through this course are the major part of the course, we will use a running example. So while we discuss about different points, we will take glimpses from one running example about a leave management system of an organization. So very soon we will introduce you to the general notion of the system, what does an organization think about leave management, what is required and then we will repeatedly use that side to side, we will have examples from Booch's book.

Will have a separate running examples in the assignments that you will do that is an assignment management system where the assignments that students do are managed through that system beside we have Booch's examples from Booch's book and for complete workout we will have couple of different example systems or postal management story management in a newspaper house car management for a rental car company and some of the Booch's OOAD examples.

(Refer Slide Time: 27:57)


Instructor and TAs

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

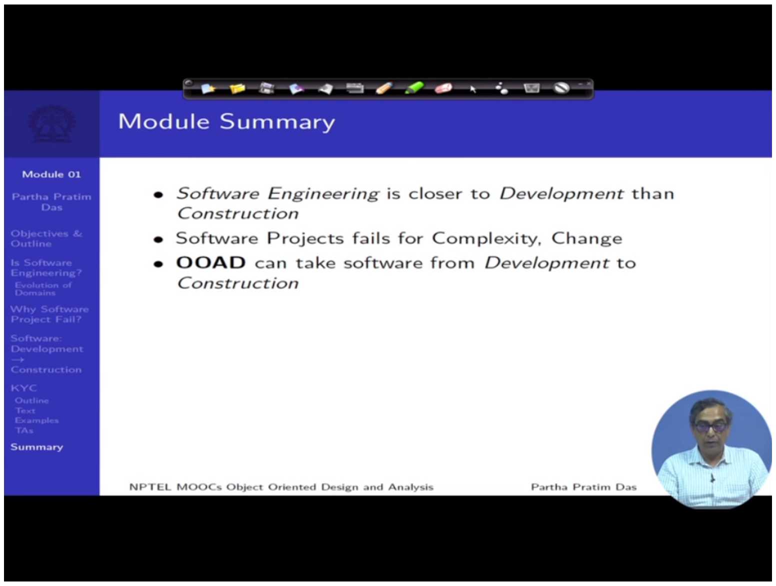
Summary

Name	Mail	Mobile
Partha Pratim Das, *Instructor*	ppd@cse.iitkgp.ernet.in	9830030880
Tanwi Mallick, *TA*	tanwimallick@gmail.com	9674277774
Srijoni Majumdar, *TA*	majumdarsrijoni@gmail.com	9674474267
Himadri B G S Bhuyan, *TA*	himadribhuyan@gmail.com	9438911655

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 19

Here am your instructor as you know so my contact is given here and am assisted by 3, Tanwi Mallick, Srijoni Majumdar and Himadri Bhuyan. So they will be accessible on these emails and phone numbers in case you need.

(Refer Slide Time: 28:18)


Module Summary

Module 01

Partha Pratim Das

Objectives & Outline

Is Software Engineering?

Evolution of Domains

Why Software Project Fail?

Software: Development → Construction

KYC

Outline

Text

Examples

TAs

Summary

- *Software Engineering* is closer to *Development* than *Construction*
- Software Projects fails for Complexity, Change
- **OOAD** can take software from *Development* to *Construction*

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

So we will conclude, summarize that software development is closer to development, that is than to construction. This is what we have seen through the simple discussion. We have also observed that software projects fails due to complexity due to regular need of change and we have noted and this is what we will demonstrate through the course that OOAD will take software from development to construction over a period of time.