Software Engineering Capstone Introduction

Lecturer: Ngo Huy Bien
Software Engineering Department
Faculty of Information Technology
VNUHCM - University of Science
Ho Chi Minh City, Vietnam
nhbien@fit.hcmus.edu.vn

Objectives

- To present *what* is software engineering capstone.
- To present *why* software engineering capstone is important.
- To create *artifacts* for each software engineering stage.
- To apply *best practices* for each software engineering stage.

Contents

I. Software development jobs
II. Software development quick-start

References

1. Steve McConnell (2004). Code Complete. Microsoft Press.
2. Craig Larman (2004). Applying UML and Patterns. 3rd Edition. Prentice Hall.

How to Get a Job?

Java Developer Hire ($600 - $1000)

- Education: Bachelor of IT/Software Engineering or Information System.
- Knowledge of OOP, UML and MVC concept.
- Programming languages/ technologies – Java 8, Java EE, EJB, JPA, Hibernate, Spring, JUnit, Mockito, Eclipse RCP, WebServices (RESTful, SOAP), JavaScript, Cordova, JSF, Angular (2), HTML5, CSS3, JQuery, etc.
- Knowledge of Database and have done through one of the database management system Oracle, SQL Server, MySQL.
- Familiar with quality assurance activities (CMMI)

- Develop effective solutions for complex business problems in a variety of customer projects
- Integrate new components and technologies into existing software systems.
- Conduct system documentation.
- Provide system maintenance service.

PHP Developer Hire ($600 - $1000)

- Education: Bachelor of IT/Software Engineering or Information System.
- Being expert in PHP, Javascript, HTML, CSS, SQL Databases (preferably MySQL), AJAX.
- Having solid competency in all aspects of LAMP stack
- Having experience with SVN or GIT
- Having worked with at least one PHP framework (e.g.: CakePHP, Laravel, CodeIngiter, Drupal or any MVC framework)
- Having worked with at least one Javascript library (eg.: JQuery, Bootstrap, dojo, ...)
- Having sound knowledge in web-application security, debugging and tuning web-applications
- Having experience in agile software development, i.e. Scrum

- Understand and interpret (in writing) project functional requirements.
- Outline work task breakdowns for programming side.
- Develop web applications.
- Coordinate and participate in peer code review

NodeJS Developer Hire ($500 - $1200)

- Degree in Computer Science, Information Technology or equivalent.
- Experience with ReactJS, NodeJS, Golang, Docker.
- Strong proficiency in JavaScript, including DOM manipulation, the JavaScript object model, specifications of ECMAScript.
- Experience with common front-end development tools such as Babel, Webpack, NPM,...
- Knowledge of modern authorization mechanisms, such as JSON Web Token.
- Understanding of how applications interact with the systems and have experiences with modern software engineering practices and paradigms.
- Experience in software development on Linux/Unix, webpage optimization and web browser technology environment would be a plus point.
- Working proficiency and communication skills in verbal and written English.

.NET Developer Hire ($400 - $1500)

- Education: Bachelor of IT/Software Engineering or Information System.
- Strong knowledge and experience in system design and Web programming with Microsoft .NET and other Microsoft technologies (C#, ASP.NET, XML Web Services, ADO.NET, WCF, Linq, XAML, ASP.NET MVC, Web API, Entity Framework...)
- Having experience with database design and programming (MS SQL Server, or Oracle, MySQL,...)
- Having worked with SVN, MS Team System, IIS,...
- Familiar with software development process.
- Good soft skills: team working, problem solving, presentation...

- Provide technical & analytic support to customers to identify requirements and develop qualified functionaries/projects.
- Outline work task breakdowns for programming side.
- Participate in system design and implementation.
- Coordinate and participate in code review.

Mobile Developer Hire ($1000 - $1500)

- A University Qualification in Software Engineering or a related degree with at least 02 years experiences in.
- Good knowledge in OOP, design pattern, unit testing.
- Native Android or iOS: Mobile app development and deployment on Android or iOS platforms.
- Java SDK or Android SDK
- C/C++, Swift/Objective C, iOS SDK.
- React Native Android/iOS: React JS, React Native, Redux, ES6.
- Experience in Databases like – Oracle / MySQL, NoSQL Databases (Mongo, etc.)
- Honest, energetic and cooperative in team.
- Problem solving, punctual.

- Work directly, in English, with the founders of new and rapidly growing Internet ventures to understand their vision and implement it on the web and on mobile platforms.
- Use eXtreme Programming (XP) and Scrum agile development methods which include paired programming, Behavior Driven Development (BDD), Test Driven Development (TDD), and emergent business-value prioritized development.
- Develop applications on Cloud platforms that can grow to support millions of users.

Embedded Developer Hire ($600 - $1200)

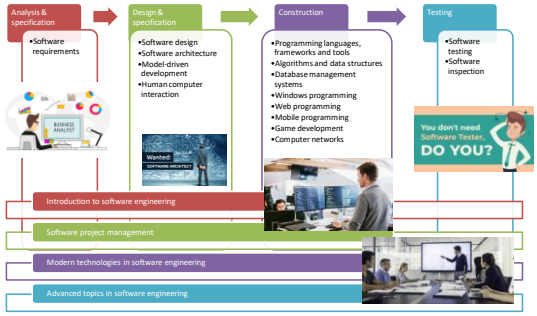
- Bachelor degree or above in related area;
- Development skill: Design, Coding, Testing, Debugging;
- Embedded C/C++ development;
- 1 – 4 years of working experience in Automotive related fields;
- Have experience in: Linux OS, Cygwin;
- Knowledge in automotive communication bus (CAN, LIN, Flex-ray), micro controller;
- Knowledge in AUTOSAR;
- Familiarity with SW development tools: emulators, debuggers, CANoe/ CANalyzer etc.
- Experience in formal software engineering methodology and process (CMMI, A-SPICE)
- Able to learn independently and work well under tight development schedules;
- Able to explain complex technical information clearly and succinctly;

Machine Learning/AI Developer (Data Scientist) Hire ($1000 - $2000)

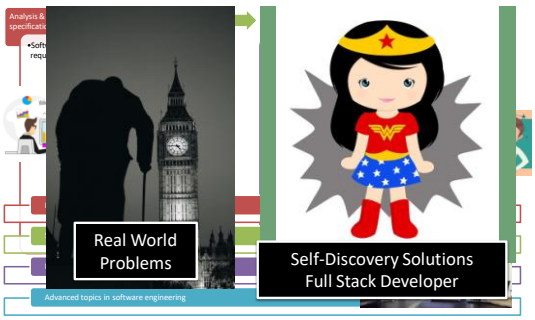
- A University Qualification in Mathematics or Computer Science or Computer Engineering.
- Experience using statistical computer languages (R, Python, SLQ, etc.) to manipulate data and draw insights from large data sets.
- At least 2 years' experience developing projects related to machine learning, chatbot, AI or data mining.
- Demonstrated ability in selecting, developing, and applying machine learning and data mining techniques (clustering, decision tree learning, artificial neural networks, etc.).
- Experience dealing with large data sets.
- Experience in chat bot development or machine learning tools (such as Tensorflow), NLP technologies or AI related projects.
- Knowledge of advanced statistical techniques and concepts (regression, properties of distributions, statistical tests and proper usage, etc.)

Prerequisite: Creating a Resume

What You Learned



Why Software Engineering Capstone?



Prerequisites: Basic Know-How

- *Domain knowledge (Concepts & business processes)*
- Development process (RUP, Scrum, Kanban)
- Planning (Estimation, scheduling & status reporting)
- Requirements (Elicitation, analysis & modeling techniques)
- User interface designs (Best practices & tools)
- Architecture & designs (Data structures, algorithms, OOAD & design patterns)
- Coding & testing (Programming languages, libraries, frameworks and tools)
- Infrastructure (Cloud services, OS & tools)

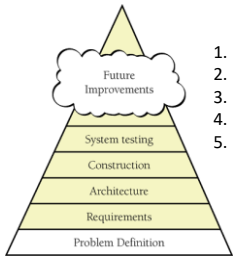
Project Idea

- Group number: X (1 line)
- Project name: ABC (1 line)
- *Problems*, pain points (NOT features): (3 lines)
- Solution: 3 lines or high-level architecture
- Core features: maximum 5 features
- *2 business cases (real world scenarios or workflows)*.
- Competitors (minimum 3): Names, their strengths and weakness.
- Your differentiators.

7 members
8 hours per week / member

Where Do We Start?

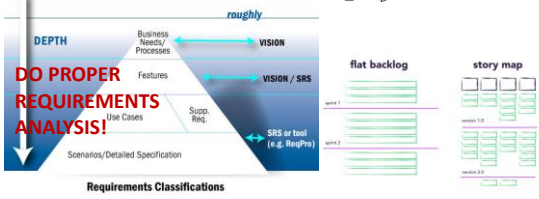
Problem-Definition [1]



1. What is the problem?
2. What is the *need*?
3. Who *really* needs your solution?
4. What are the *business scenarios*?
5. What is the *evaluator pitch*?

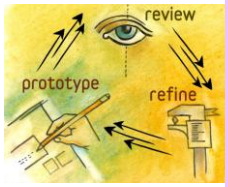
Requirements

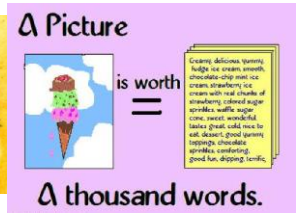
1. What are the *features*?
2. What are the *use cases/user stories*?
3. Who are the *actors*?
4. What are the *inputs*?
5. What are the *outputs*?



Prototyping

1. How will a real world *problem* be solved?
2. How does a use case/story map *look like*?
3. How *fast* does a use case/story map need to satisfy?





Rough Unreliable Range Of Cost [2]

- Is it *$10K-100K* or in the *millions*?
- Should we proceed or stop?

Gantt Chart - Project Development

Activity Name	1st Quarter			2nd Quarter			3rd Quarter			4th Quarter		
	Jan	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec
1. Design and Development												
2. Cost Estimates												
3. Construction												
4. Testing												
5. Celebrating												★

Possible Inception Artifacts

- Glossary
- Executive summary
- *Project vision*
  - *Scenarios*
  - *Domain models*
- *Use cases / User stories / Prototypes*
- *Proof-of-concepts*
  - *Architecture*
  - *Tools*
- Risk list & Risk management plan
- Development method & rough estimate

Isn't that *A LOT* of documentation?

Misunderstanding

- It is more than *"a few" weeks* long for most projects.
- There is an attempt to define *most of the requirements*.
- Estimates or plans are expected to be *reliable*.
- There are *NO* Scenarios or Domain Models.
- *All* the use cases were written in detail.
- *None* of the use cases were written in detail.

Important Notes

- Defining the vision and obtaining an order-of-magnitude (unreliable) estimate requires doing some *requirements exploration*.
- However, the purpose of this phase is *NOT* to define all the requirements, or generate a believable estimate or project plan.
- **Again, Scenarios, Domain Models and Use Cases (User Stories)!**

How to Evaluate the Artifacts?

Why is evaluation in this phase *the most difficult*?

Wrong Estimate and Plan!!!

Our estimate is *not reliable*.
We are stuck and produce *nothing useful*.

How Can We Fix It?

Possible Analysis & Design Artifacts

- Process flow / Activity diagram
- Entity relationship models
- User interfaces
- Class diagrams
- Sequence diagrams
- Algorithm flowcharts

Didn't all that analysis, modeling and design take *WEEKS* to do?

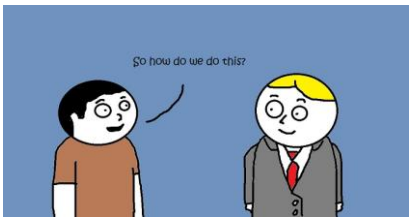
How to Evaluate Analysis & Design Artifacts

How do I know whether my ERD is *correct*?

Important Notes

- *Early programming*, tests, and demos help provoke the inevitable changes early on.

How to Code the First Feature?



Tools and Method Selection

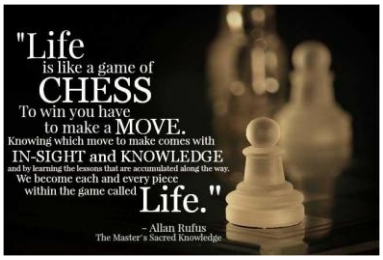
How about architecture & design?

Possible Development Artifacts

- Update architecture.
- Updated entity relationship models.
- Updated user interfaces.
- Updated class diagrams.
- Updated sequence diagrams.
- Updated algorithm flowcharts.
- *AND SOURCE CODE.*

& Learning & Problem Solving

Insight and Knowledge



Important Notes

- *Again! Early programming*, tests, and demos help provoke the inevitable changes early on.

Thank You & See You Again