

**Notes from Course lecture:**

Lecture 13 Partitioning:

- This course lecture addresses the following. Please refer Course lecture for details. For each point, presentation slide title and time are provided within brackets.
  - Partitioning - Divide and conquer. (3 Subtopics, **00:37** and 4 Divide et impera\*, **00:46**)
    - Divide system and conquer complexity
  - Systems can be partitioned by functions. (5 By Function, **01:44**)
  - Systems can also be partitioned by geographic location. (6 By Geographic Location, **05:08**)
  - Systems are partitioned by country of origin. System needs to be integrated seamlessly. (7 By Country of Origin, **07:33**)
  - Systems are partitioned by security level. Security can be classified top secret, secret and unclassified. (8 and 9 By Security Level, **12:12** and **16:27**)
  - Partitioning by developing contractor. Partitioning by team / subcontractors instead of country. Different team agrees and establishes contract before bidding. (10 By Developing Contractor, **17:46**)
  - Government or country picks the prime contract team and each contract team picks their team. (11 By Prime Contractor Team).
  - Partitioning can be done by technology. Hardware, speed, etc. (12 By Technology, **22:09**)
  - Partitioning by dissimilarity. Example using dissimilar microprocessors to avoid failure on all microprocessor. (13 and 14 By Dissimilarity, **23:01** and **25:15**)
  - Partitioning by federates. Federate is a member of a high level architecture federation. All applications participating in a federation are called federates. Federation is a named set of interacting federates, a common federation object model, a supporting run time interface, that are used as a whole to achieve some specific objective. (15 By Federates, **26:00**)
  - Partitioning can be done based on different layer like application, presentation, session, network, etc. Each layer uses the layer immediately below it and provides a service to the layer above. (16 By Layer, **33:52**)
  - Partitioning of system can be done based on software language. Example Java for middleware, C++ for framework and algorithms. (17 By Software Language, **37:17**)
  - Partitioning of the system can be done for survivability with duplication at highest level. (18 By Duplication for Survivability, **41:54**)
  - Partitioning by modes and states. For example training mode or operational mode or maintenance mode or by state like weapons loose, weapons free, etc. (19 By Modes and States, **42:23**)
  - Partitioning is done based on operational stage. Example pre-launch, boost, glide and kill (20 By Operational Stage, **46:32**)
  - Based on different build. Between development and operation the system will built incrementally (21 By Build Stage, **47:30**)

Lecture 13R Sandstorm:

- This course lecture addresses the following. Please refer Course lecture for details. For each point, presentation slide title and time are provided within brackets.
  - Sandstorm has three major system – perception, autonomous and mobility. (3 Sandstorm, **01:15**)
  - Autonomous hardware includes XEONs, Itanium II, power control, networking, etc. (4 Autonomous Navigation System, **02:02**)
  - Perception hardware includes SICK LIDAR, Single Axes Gimbal, Stereo Vision Cameras, etc. (5 Perception H/W locations, **04:43**)
  - Sandstorm system flow of information is addressed in the picture presented in 6 and 7 Sandstorm System, **07:55** and **12:02** as well as 9 and 10 Sandstorm System, **16:27** and **18:43**.
  - Mobility platform of the system – driven by wire. 11, 12 and 13 Drive By Wire, **19:02**, **22:55** and **24:41**.
  - Network associated with Sandstorm which are lot of them. 14 Autonomous Navigation Interfaces, **25:01**.
  - Details of how data provided by different system for autonomous navigation system to control the vehicle. 15

**Notes: (Record key insights from readings and discussions.)**

- and 16 Perception Data to Information, **31:27** and **33:35** as well as **18 and 19 at 37:10 and 39:57**.
- Shuttle Cost Pyramid express the percentage cost by different field like engineering, technicians, etc. (**20 Shuttle Cost Pyramid, 40:28 and 21, 23 and 24 Current Shuttle Environment, 41:28, 43:37 and 45:32**)
- Data owned and consumed by different portion of the system is addressed. (**25 ACCMS Top Level Architecture, 46:28**)
- Web Services address the interaction (**26 and 27 Integration via Web Services, 47:07 and 48:07**)
- System Data Life Cycle is addressed at slide **28, 48:41**.

**Lecture 14 System Architecture with Examples:**

- This course lecture addresses the following. Please refer Course lecture for details. For each point, presentation slide title and time are provided within brackets.
- Architecture is the manner in which the components of a computer or computer system are organized and integrated. (**4 Architecture Definitions, 01:14**)
- Architecture is the extensibility layer of a system design. The architecture specifies requirements for present and future implementation. (**5 My Thoughts, 01:40**)
- Architecture Perspectives (**6 Architecture Perspectives, 05:34**)
  - Conceptual architecture describes the system in terms of its major design elements and the relationships among them.
  - Module interconnection architecture encompasses two orthogonal structures.
  - Execution architecture describes the dynamic structure
  - Code architecture describes how the source code, binaries and libraries are organized.
  - <http://www.sei.cmu.edu/architecture/definitions.html>
- Structural, Framework, Dynamic, Process, Security and Safety are different architectural models. (**7 Architectural Models, 10:57**)
- Architectural types can be classified as technical, operational and systems. (**8 Architectural Types, 11:51**).
- The Architecture Based Design (ABD) Method provides structure in producing the conceptual architecture of a system. It describes the system(s) being designed in terms of the major design elements and the relationships among them. The conceptual architecture represents the first design choices made during a development process. (**9 The Architecture Based Design Method, 12:44**)
- ABD method depends on determining the architectural drivers for a system. ABD has three foundations. (**10 ABD Drivers and Foundation, 13:12**)
  - Functional decomposition.
  - Realization of quality and business requirements through the choice of architectural style.
  - Software templates.
- Software templates define what it means to be a software element of a particular type. It includes patterns that describe how all elements of this type must interact with shared services and the infrastructure. (**11 Software Templates, 16:52**)
- Designing architecture for a product line or long-lived system is difficult because detailed requirements are not known in advance. (**12 and 13 ABD Wisdom, 22:06 and 25:31**)
- Microplex – A plexus of microprocessors (from **14 until 21 Example Architecture – Microplex, 27:48 until 42:03**)
- Second example. (from **22 until 27 Example Architecture – Microsoft .NET, 42:03 until 58:56**)
- Application Service Program is similar to “Time Sharing”. Outsourcing information processing service. (**28 ASP, 58:56**)
- Instant messaging, Web conferencing and Voice over TCP/IP are some of current architectural drivers. (**29 Some Current Architectural Drivers, 1:08:52**)

**Lecture 14R Red Team Mapping System Architecture:**

- This course lecture addresses the following. Please refer Course lecture for details. For each point, presentation slide title and time are provided within brackets.

### **Notes: (Record key insights from readings and discussions.)**

- Red Team Organization Chart addresses different divisions with department leads. (2 Red Team Organization Chart, 00:19)
- Mapping is extremely important. 250 miles. Developing great maps is important to accomplishing the task. Handling the vehicle a best planned route on the race day. (3 Importance of Mapping, 01:37)
- Mapping data is important. But why group of people need to work on it? Best available data is 10m horizontal accuracy and 30m vertical accuracy. That is not enough. (4, 5 and 6 What's so tough about that?, 04:23, 06:04 and 06:31)
- Time, Money and Hardware are required to solve this problem. (7, Expenses to solve the problem, 08:14)
- The issues are addressed by breaking down the requirement. Software, Hardware and People (9 Mapping as a system, 09:19)
- Slide 10, 11 and 12 Hardware at 10:28, 12:48 and 14:23 address the hardware requirement.
- With respect to software, map editing software is important one. Data maintenance is another. The data needs to be automatically backed up. (13, 14 and 15 Software at 15:36, 17:03 and 17:32)
- The last one is people. Need to improve map and there is no way to automate the process. Team needs to be motivated. Many people are involved. Training and quality check are adapted. (16, 17, 18 and 19 People at 18:13, 18:27, 19:14 and 19:50)
- Red team faced couples of major problems like hard drive failures, change in race rules, registration errors in images and absurdly edited maps. (20 Major Problems Faced, 21:14)
- Perfect maps leads to perfect route. (21 System Diagram, 23:06)
- The Red Team Organization was re-organized based on the importance that needs to be given to route. (22 Revised Red Team Organization Chart, 24:25)

### **Notes from Professor lecture:**

(Time and slide information are not captured since the video time is short)

#### **Week 10:**

- There is no single template that can be used to define architecture.
- Two Architecture reading sets provide a starting point.
- Architecture needs to be prepared after doing research, after understanding about stakeholders, what the firm use to produce system architecture, etc.
- Architecture documents are produced after understanding about the audience of the document. Who will read and use the document.
- Architecture document can be used by different audience for different purpose. The document can be used for planning, design, risk management, etc. Understanding about audience help to prepare the document appropriately.
- Utility tree is the starting point. It helps establish the set of architecturally significant requirements that needs to be addressed.
- The Pareto Critical Few – Again there are lot of things that can be considered while preparing the architectural documents. What are important needs to be addressed?
- Review guidance, double check team deliverable and ask question to professor whenever required.

### **Notes from Reading: (Architecture Set 2)**

- Independent Verification and Validation – John Du Vu, Chief Engineer and Technical Fellow Boeing Company. (4 pages)
  - Verification is a process to evaluate software to determine whether the products of each phase full-fill all the requirements of the previous phase - page 1.
  - Validation is to evaluate software product to determine whether a product or product component fulfills all requirements when placed in its intended environment - page 1.
  - Independent reviews have the following benefits – page 1.
    - Detect errors early.

**Notes: (Record key insights from readings and discussions.)**

- Establish traceability.
    - Confirm requirements are consistent, unambiguous and testable.
    - Ensure intended functions are implemented correctly.
    - Risk assessment.
    - Provide independent view of process and product quality and development status.
  - Independent Verification and Validation starts from systems requirement. Developer creates software requirements based on system requirement followed by software design and software code and test to produce software product. Verifications can be done during requirements, design and code. Validation happens on a software product to confirm system requirements are met. Refer page 2 for detail.
  - As per IV & V requirements are analyzed for operational need, validated for system specification and verified for software requirements. Refer page 2 for additional details.
  - IV & V Requirements Analysis Techniques involve
    - Decomposition
    - Independent derivation
    - Trade-Off analysis
    - Hierarchy or criticality determination
    - Requirements synthesis and
    - Report generation.
    - Refer page 3 for additional information.
  - IV & V code analysis involve
    - Internal review by peer
    - Code walk-through by group and
    - Code inspection by independent team.
    - Refer page 3 for additional detail.
  - Review and inspection provide benefits. The benefits helps in reducing re-work. The amount of benefit depends upon whether the review process was conducted. Example the review process can happen on the design or code or test phase. Page 4 for additional detail.
  - IV & V test monitoring involves test planning, defining procedure, monitoring and compliance. This helps to confirm whether the results are satisfactory or any the deficiencies are identified. Page 4 for additional detail.
- New Testing Model – New Models for Test Development by Brain Marick, Testing Foundations. (13 pages)
    - The paper address why the quite popular ‘V model’ is a bad model and describes about a model which the author thinks is better. (page 1)
    - V model describes software development in phases like requirements, specification, design, code, etc. Typical waterfall model which has lot of problems during development. When testing activities are added to the model, refer page 2 for cross reference between development phase and testing phase. Example unit testing, followed by integration, system and acceptance testing.
    - Test designs are done when the appropriate development documents are prepared. Refer page 3 for how design test document activities are mapped to development activities.
    - V model suggest running each unit test before running the integration test. The test cases are decided based on interfaces. But the unit cannot be tested in isolation which the author points out as a bad idea. Author suggests allowing testers to consider the possible savings of deferring test execution. Refer page 4.
    - As per author a test is designed to run to find bugs through unit test or part of subsystem along with tests designed to find integration problems. With that the distinction between ‘unit’, ‘integration’ and ‘system’ tests begins to break down. Refer picture in page 5 to understand how they are related.
    - When test design is derived from a description of a component of the system, the model must allow such tests to be executed before the component is fully assembled. Refer to picture in page 6 to understand how they are connected.
    - According to author, the V model fails because it divides system development into phases with firm boundaries between them. It discourages people from carrying testing information across those boundaries.

**Notes: (Record key insights from readings and discussions.)**

- *The model must allow individual test to be designed using information combined from various sources. The model must allow tests to be redesigned as new sources of information.*
- **Different model**
  - *A test model should force a testing reaction to every code handoff in the project.*
  - Refer page 7 to 10 for example.
  - The test effort may be degraded by poor or late project documentation, but it should not be blocked entirely.
  - *The test model must explicitly encourage the use of sources of information other than project documentation during test design.*
  - *The test model must include feedback loops so that test design takes into account what's learned by running tests.*
  - *The model must require the test planner to take explicit, accountable action in response to dropped handoffs, new handoffs, and changes to the contents of handoffs.*

**Other links:**

- <http://www.advancedscadaandtelemetry.net>
- <http://www.microsoft.com/usa/presentations/search.asp?district=gulfstates>
- <http://www.hoise.com/vmw/99/articles/vmw/LV-VM-04-99-22.html>
- <https://apps01.stricom.army.mil/applications/STOCportal.nsf>