**TERMS AND CONCEPTS**

**“V” model-** The first test designed is the last one to be consumed when it is actually being carried out. Identical to water fall but more precise. **acceptance testing-**  is a test conducted to determine if the requirements of a [specification](http://en.wikipedia.org/wiki/Specification) or [contract](http://en.wikipedia.org/wiki/Contract) are met.

**agile development-** a group of [software development methods](http://en.wikipedia.org/wiki/Software_development_methodologies) based on [iterative and incremental development](http://en.wikipedia.org/wiki/Iterative_and_incremental_development), where requirements and solutions evolve through collaboration between [self-organizing](http://en.wikipedia.org/wiki/Self-organization#Self-organization_in_agile_software_development), [cross-functional teams](http://en.wikipedia.org/wiki/Cross-functional_team). It promotes adaptive planning, evolutionary development and delivery**.  
agility-** Effective (rapid and adaptive) response to change **analysis-** a design model is created and documented using architectural models, component models, object models, and sequence models.

**analysis paralysis-** refers to over-analyzing (or over-thinking) a situation, or citing sources, so that a decision or action is never taken, in effect paralyzing the outcome. **automated testing-** the use of special software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions.

**backlog-** backlog capture the dynamic flow of work from request through completion. **CHAOS Study-** results provide a global view of project statistics but do tend to have a heavier concentration on the United States and Europe. **code and fix-** simple, quick, Avoids needing to communicate / coordinate / cooperate with others. BAD- Cost overruns / schedule overruns. Low productivity / poor use of resources **critical system-** a computer system whose failure can result in significant economic, human, or environmental losses. **embedded system-** a software system embedded in a hardware device **extreme programming-** a widely used agile method of software development that includes practices such as scenario-based requirements, test-first development, and pair programming.

**feature creep-** ongoing expansion or addition of new [features](http://en.wikipedia.org/wiki/Software_feature) in a product, such as in [computer software](http://en.wikipedia.org/wiki/Computer_software).

**framework activity**- Activities directly related to defining and building the system and ultimately results in deployable software (analyze, design, code, test)

**JAD (joint application develop)-**  is a process that accelerates the design of information technology solutions

**iterative development-** the processes of specification design, programming, and testing are interleaved..

**object-oriented-** an approach to software development where the fundamental abstractions in the system are independent objects.

**open-source software-**  source code for a system is made public and external users are encouraged to participate in the development of the system **RAD (rapid application development)-** rapid delivery of software **Rational Unified Process-** a software process product, product includes a hyper linked knowledge base with sample [artifacts](http://en.wikipedia.org/wiki/Artifact_(software_development)) and detailed descriptions for many different types of activities.  **requirements engineering-** is a systems and software engineering process which covers all of the activities for a computer-based system. **regression testing-** is any type of [software testing](http://en.wikipedia.org/wiki/Software_testing) that seeks to uncover new [software bugs](http://en.wikipedia.org/wiki/Software_bug) in existing [functional](http://en.wikipedia.org/wiki/Functional_testing) and [non-functional](http://en.wikipedia.org/wiki/Non-functional_testing) areas of a system after changes **risk management-** identifying the risk, planning measures to put in place if risks arises **RUP(rational unified process)-** phases are inception, elaboration, construction, and transaction. **scrum-** commonly used for product development, especially software development. **Software Development Life Cycle-**  process of creating or altering information systems that people use to develop these systems. **SEI (software engineering institute)-** a research and technology transfer center, improve US **+spiral model-**(P) combining elements of both [design](http://en.wikipedia.org/wiki/Design) and [prototyping](http://en.wikipedia.org/wiki/Prototyping)-in-stages, in an effort to combine advantages of [top-down and bottom-up](http://en.wikipedia.org/wiki/Top-down_and_bottom-up_design) concepts. **sprint-** itemizes the actual work tasks(iterative cycle) **sprint backlog-** the “contract of work” with external stakeholders  
**stakeholder-** a person, group, who affect organization's actions (vv) **systems engineering-** (P)concerned with specifying a system, integrating comps. and test reqs.met. **TDD (test driven development)-** executable tests are written before the program code.  
**umbrella activity-** Activities that control the steps or stages of framework activities **throwaway prototypes-**  the creation of a model that will eventually be discarded. **unified process-** extensible framework which should be customized for specific organizations or projects.**unit testing-** individual units of [source code](http://en.wikipedia.org/wiki/Source_code), sets of one or more computer program modules, usage procedures, operating procedures, are fit for use.

**“throw over the wall”-**Passing on a project without any sort of consultation.

**build to schedule-**a metric used to how well a process is actually managed versus a plan or schedule **chaotic system-**act, sense, respond; no real relationshp (btwn cause and effect)  **CMMI-**a process improvement approach. Has three main items it looks at: Product and service development, Service establishment/management, Product and service acquisition **complex system-**probe, sense, respond; relationship between cause and effect can only be discovered in retrospect. emergent practice. **complicated system-**sense, analyze, respond; good practice. **computer science-**science field addressing computers **construction-**creation of an object, program, something **customer-** typically the end user, possibly represented during development by a client **cyber security-**information security in the computer world. Not necessarily limited to the internet **cycle-time-**time it takes to produce a working/cohesive unit of a project **Cynefin framework-**basically a method/perspective to describe and analyze types of systems. has four main types - simple, complicated, complex, chaotic**.  
daily stand-up meetings-**meeting type from scrum; meet for under 15 mins a day, discuss goals/progress/obstacles.  **design-**framework for the project, how you'll go about it **disorder-**a system that cannot be categorized **elaboration-**2nd phase in Unified Process: recognize all risks, solidify architecture, understand requirements.to produce the prelim version. **inception-**initial idea formulation(the very beginning) **integration testing-**individual software modules being tested together

**KISS-**keep it simple, stupid **OSS-**open source software development **planned development-**development that is charted out before development happens **prescriptive development-** structured, rigid development processes (think Waterfall). stages cannot be advanced from without approval. **project backlog-** list of features that stakeholders/devs think should be in the program. possibly prioritized by percentage of success. **prototyping-**creating early versions of an end product to improve it via user feedback **refactoring-**altering internal structure of project without changing what it does **retrospective-**looking back **scrumfall-**term referring to present, actual state of agile; structured, prescriptive processes for development framework **scrummy-**scrum in the management **simple system-**sense, categorize, respond **software engineering-**discipline of the creation and maintenance of software**SWEBOK (Software Engineering Body of Knowledge) -** everything a software engineer should technically know/be proficient at **test first-**keep design simple so its easy to write tests for **testing-** quality checks written for runin code **UP-** unified process **systems testing-** takes place on integrated whole system, evaluates systems' efficiency (rgding) requirements

Prepare to Explain

* Draw / describe a named software process model
* Describe benefits / drawbacks of prescriptive software processes
* Describe benefits / drawbacks of agile software methods
* Compare and contrast prescriptive (planned) and agile development philosophies  
   -Prescriptive code is also seen as the product of several lines of codes ready to be executed or to become performative. This layer is created as code, and becomes a signifier of productivity.
* What is "agile" versus "agility" in software development?  
  Agile is all about embracing the chance, evolution. Has all agile characteristics.  
  Agility can react & adopt to change but, not as good as agile though.