Class 5 CS540

Gregg Vesonder

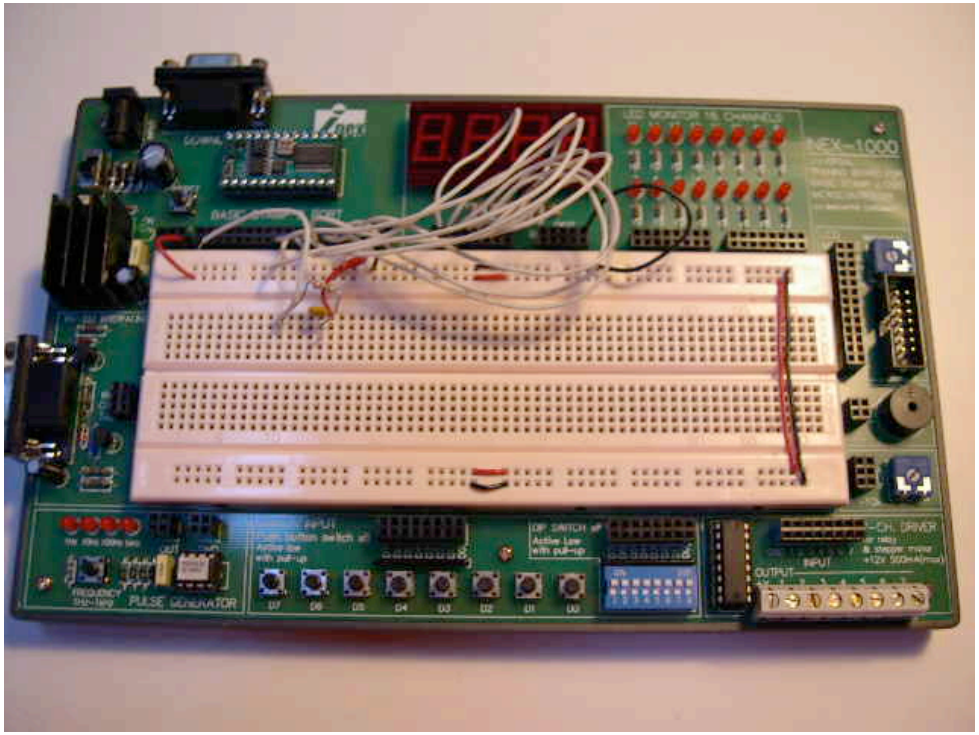
Stevens Institute of Technology

Roadmap- Class 5

- Clarifications from last class
- Log Book volunteer
- Brooks
- Software Quality Assurance
- Test
- Reading: Brooks Chapter 6, BY 393-398
- Next class - BY Chapter 11.

Last Class Clarifications

- Who's this for? Micro/macro
- Breadboards



Calendar -Key Dates

- October 3rd - first test
- Oct 11th - class on Tuesday not Monday!
- November 7th - second test
- November 21st - log books due
- December 12th - final exam

Log Book

- Meisters, extreme builders
- Volunteer

Brooks Chapter 6
On the life of a spec

- Must have specs and the changes must be quantized on scheduled dates (and source controlled)
- Be careful about overprescription in any spec -- there is always temptation.
- Spec should be done by 1 or a few -- should have a common, consistent voice
- Should say what it is and what it is not and what constraints it reflects (or imposes)
- If using formal descriptions along with prose, one must be the standard
- "never go to sea with two chronometers, take one or three."
- "If you have multiple standards, there are none!"

Conferences and Courts

- Weekly half day meetings -- I call this the heartbeat meetings
  - Includes architects, reps of implementers and reps of users
  - Same group each meeting
  - Everyone has stake - no advisers or observers
  - Proposals written by one person (agenda is distributed ahead of time and followed)
  - Decisions are discussed - creative potential
  - Architect is the final say

Supreme Court Sessions

- Semiannual for two weeks (clearly this should be adjusted by size and need)
- Should not be ignored, "bonding of the decisions"
  - Resolves issues of the team and dissatisfaction over some issues
  - Hold before major freeze dates of spec
  - "not only for resolving decisions but getting them accepted" … everyone was heard

Nits from Chapter 6

- Nice thing to have in specs is the declaration of passed parameters
- Do not overprescribe especially regarding implementation
- The telephone log -informal interactions between implementers and architects -- log and distribute them
- Wiki's
- The tester as your best friend, link between designer and implementer
- Test early and often

Wiki

- A web site on which anyone can contribute pages, a wiki page also can be edited by anyone
- Produced by collaborative software called wiki
- WikiWiki comes from Hawaiian meaning quick or superfast
- Wiki pages have very simple markup and can be edited in web pages
- Links are created using a link pattern initially CamelCase (e.g.TableOfContents)
- Software available on web - google WikiWikiWeb, the 1st wiki.

On managing software quality

Q

Peters & Waterman, In Search of Excellence, a key factor of successful companies

On Quality

- The cost of low quality ranges from dissatisfied customers to costing lives.
- Premise is that the quality of a product is largely based on the Quality of the process that leads to the product.
- We should strive to numerically describe quality - the other Q - Quantitative, "from response time is very fast to an average response time of < 1 second with no response taking over 3 seconds."
- What Quality is differs by audience - each (tester, end user, administrator) has their own perspective.

Approaches to Quality

	Conform	Improve
Product	ISO 9126	Best Practices
Process	ISO 9001 SQA	CMM SPICE Bootstrap

Approaches to Quality -2

- TQM, Total Quality Management, emphasizes the eclectic view, the pursuit of excellence in everything
- SQA - sees to it that the work is done the way it should be done
- CMM (and SPICE and Bootstrap) improves the development process or the process to improve the development process

Regress on Measurement

(Kitchenham, et.al.)

- Entity - an object in the "real" world, where real contains projects and software
- Attribute - entities have properties (many times more than 1)
- Attribute relation - different properties can be related, e.g, effort and cost
  - An attribute of entity X can be related to an attribute of entity Y, e.g., experience of programming time and project cost
- Attribute-relation model - relationships can be expressed in a formal model, model predicts values of an attribute by using other values in the model.
- Measurement is mapping from real world to formal, relational world, measure is the number or symbol assigned to the attribute

On Measurement

- Value- each attribute is assigned one
- Unit - the value is expressed in a unit, dollars, month, …
- Scale type: unit properties
  - Nominal - classification
  - Ordinal- linear ordering, shortest to tallest, values are not necessarily equidistant
  - Interval - distance between successive values the same
  - Ratio - there exists a value of 0, age
  - Absolute, # of occurrences, number of errors
- Each scale type permits certain operations -- ordinal only median not average

Measures -3

- A valid measure is one that satisfies the representation condition - it preserves empirical relations between objects in the real world, if A is observed to be faster than B then the relation should reflect it.
- This exercise is important to stress care in measurement - one example is that McCabe's metric does not necessarily preserve the representation condition in all instances -- higher complexity numbers in McCabe do not always mean higher complexity, intelligence tests do not measure intelligence. But … it is all in the concept and the pragmatics

On Quality

- Quality is tough to measure but easy to recognize (and there is agreement, inter rater reliability)
- IEEE defines it as "the degree to which a system, computation or process meets customer or user needs and expectations"
- Perspectives:
  - User: degree to which requirements are met: correctness, reliability, usability, …
  - Customer: factors relating to the structure of the system: maintainability, testability, portability

McCall's taxonomies

- (taxonomies are common in the early stages of a science)
- High level Q Factors, external attributes that are measured indirectly and Q Criteria measure subjectively or hopefully objectively. By combining Q Criteria one measures the Q Factor. Subjective measures can be broken down even further to find an objective base
- 3 classes of Q Factors: operation, revision and transition
- Q factors are not independent

McCall Examples

- Operation: Correctness (completeness & consistency), Usability (communicativeness, operability & training)
- Revision: Maintainability(conciseness, consistency, modularity, self-documentation, simplicity)
- Transition: Portability(hardware independence, modularity, self-documentation, software system independence)

ISO 9126

- Another attempt to define Quality characteristics and subcharacteristics
- Addresses product not process
- Only subcharacteristics visible to user
- Provides metrics for each subcharacteristic
- The current answer - develop your own based on these and relevant to your needs but remember that quality requirements that cannot be measured cannot be controlled

ISO 9126

- Characteristics and sample subcharacteristics:
  - Functionality (suitability, security)
  - Reliability (fault tolerance)
  - Usability (understandability)
  - Efficiency (resource use)
  - Maintainability (changeability)
  - Portability (replaceability)

Garvin on Quality

- 5 perspectives/definitions of quality:
  - Transcendent, innate excellence, obvious
  - User based, fitness for use, addresses needs (acceptance test)
  - Product based quality as in ISO 9126
  - Manufacturing based - conformance to specs
  - Value based - show me the money

TQM- Total Quality Management

- Stresses improvement rather than conformance, CMM builds on TQM
- 3 principles:
  - Customer value strategy - benefits vs sacrifices caused by the product
  - Organizational system - eliminate complexity not people, people are a critical resource and focuses on how software melds with organizational practices
  - Continuous improvement - proactively based rather than reactively based.

ISO Quality System

- ISO set up a series of standards for quality management
- ISO 9001 most suited for software - model for quality assurance in design, development, production, installation and servicing
- ISO 9004-1 contains guidelines for individual elements of various standards
- ISO 9000 process includes third party auditor, with audits every 6 months and reregistration every 3 years - expensive
- Necessary for some customers

Software Quality Assurance

- Basic idea: improve quality by monitoring software and its development process
  - Ensure compliance with established standards
  - Ensure that inadequacies are brought to the managements attention and fixed
  - They review and audit and must be separate from production
  - Support of management, have go/no go authority
  - Must be technically competent
- IEEE 730 provides a framework for Quality Assurance plan
- IEEE 983 is a complement to 730 and offers further guidelines including implementation, evaluation and modification.

IEEE 730

- Purpose
- Reference documents
- Management
- Documentation
- Standards, practices, conventions and metrics
- Reviews and audits
- Test
- Problem reporting and corrective action
- Tools, techniques and methodologies
- Code control
- Media control
- Supplier control
- Records collection, maintenance and retention
- Training
- Risk management

QAW

- Quality Attributes Workshop - facilitated method engaging stakeholders early to discover driving Quality attributes of a software intensive system
  - Results in creation of prioritized and refined scenarios
  - Provides description of Quality requirements before architecture is developed
  - It is system centric and stakeholder focused, done before software architecture is created
- Critical Quality attribute must be articulated and well understood early so it influences architecture - move to the left!
- Quality attribute examples: security, reliability, modifiability, performance, interoperability, portability

View of Traditional System Development

- Operational Descriptions
- High Lev Functional Requirements
  - Legacy Systems
  - New Systems
- A Miracle Occurs - Quality attributes are often missing from requirements document or, at best, vaguely understood and described
- Specific System Architecture
- Software Architecture
- Detailed design
- Implementation

QAW-2

- Motivation is to not rely on the miracle but clearly articulate what is needed by scenarios describing the stimulus, describes agent or factor that initiates system to react and a response, the systems reaction to the stimulus. Including the environment, the context (e.g. peak load, normal operation, maintenance mode).

Steps of QAW

- Step 1 - QAW presentation - description and participant introduction
  - 5 to 30 stakeholders
    - Stakeholders = usual suspects = end users, installers, administrators, trainers, architects, system and software engineers.
- Step 2 - Stakeholders present systems business/mission context, including high level requirements, constraints and identified Quality attributes
- Step 3 - Architecture Plan presentation, notional, preliminary architecture describing how requirements will be satisfied, key technical requirements and constraints and description of the system environment

Steps of QAW-2

- Step 4 - Identification of Architectural Drivers - facilitators summarize these from presentations in Steps 2&3, ask stakeholders for clarification, additions & deletions. Results in final list of Quality attributes to drive scenario stage.
- Step 5 - Scenario Brainstorming, stakeholders generate scenarios, each stakeholder contributes 2, facilitators assure at least one scenario for each Quality attribute driver of Step 4. The Quality attributes are operationally defined by these scenarios, hopefully, avoiding ambiguity of vocabulary.

Steps of QAW-3

- Step 6 - Scenario Consolidation, similar scenarios are consolidated
- Step 7 - Scenario Prioritization, each stakeholder has N votes (N = 30% of # of scenarios), 2 passes, 1/2 of votes on each pass
- Step 8 - Scenario Refinement, work hard on clarifying descriptions of highly rated scenarios, including describing business/mission goals affected by scenario and describing relevant Quality attributes

Example Scenario

- Scenario - when a garage door opener senses object in door's path, stops door in less than 1 msec.
- Business Goals - safest system; feature rich product
- Quality Attributes: safety, performance
- Stimulus - object in path of garage door
- Stimulus Source - object external to system, bicycle
- Environment - garage door is closing
- Artifact - system motion sensor & motion control software
- Response measure - 1 msec
- Questions - How large must an object be before detected
- Issues - train installers to prevent malfunctions

QAW Benefits

- Increased stakeholder communication
- Informed basis for architectural decisions
- Improved architectural documentation
- Support for analysis and testing throughout life of the system

CMM and others

- CMM - been there done that
- BOOTSTRAP - separate maturity rating for each of its practices
- SPICE (ISO/IEC 15504) - international initiative Software Process Improvement and Capability dEtermination

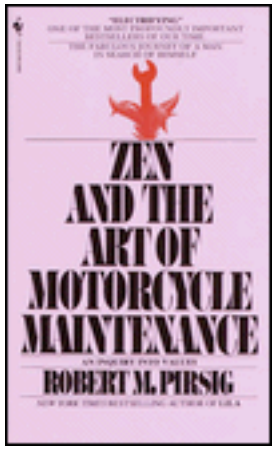
Reality Check

- The business is software, danger of a shift from developing software to developing processes, but …

- Quality is recognizable

Pirsig

- Zen and the Art of Motorcycle Maintenance: An Inquiry Into Human Values, Bantam Books, 1974. ISBN:0-535-27747-2



Thought Problem

- Your company does not have a Quality program, lately there have been some issues with software Quality and your Director has asked you to institute a program for your company. What is your plan?

So Far

- Software Process Models, Software Project Planning (woosh!), Requirements, Estimation, Risk Analysis, Multics case study, Architecture Reviews, Questionnaire Design, Software Quality Assurance
- Next Time: Configuration Management and Testing

References

- Barbacci, et.al., "Quality Attribute Workshops (QAWs), Third Edition, CMU/SEI-2003-TR-016.