Lecture 29
December 3, 2021

Life Cycle for Building Secure, Trusted Systems

- Life cycle process establish discipline, control in the building of a product or system
  - This provides confidence in consistency, quality of resulting system
- Assurance *requires* life cycle model end engineering process in *every* situation
  - Size and complexity will vary
- Life cycle defined in stages

Generic Life Cycle Model

These are present in all models, but the emphasis and focus is different for each project, and will be more detailed than what is presented here

- Conception
- Manufacture
- Deployment
- Fielded Product Life

Conception

- Idea
  - Decisions to pursue it
- Proof of concept
  - See if idea has merit
- High-level requirements analysis
  - What does “secure” mean for this concept?
  - Is it possible for this concept to meet this meaning of security?
  - Is the organization willing to support the additional resources required to make this concept meet this meaning of security?
- Identify threats, assumptions

Manufacture

- Develop detailed plans for each group involved
  - May depend on use; internal product requires no sales
- Implement the plans to create entity
  - Includes decisions whether to proceed, for example due to market needs
  - Software development, engineering process is in this stage

Deployment

- Delivery
  - Assure that correct masters are delivered to production and protected
  - Assure integrity of what is delivered to customers, sales organizations
- Installation and configuration
  - Ensure product works appropriately for specific environment into which it is installed
  - Service people know security procedures
- Example of configuration failure
  - 2013: Target breached via a third party vendor, as network architected with improper security controls

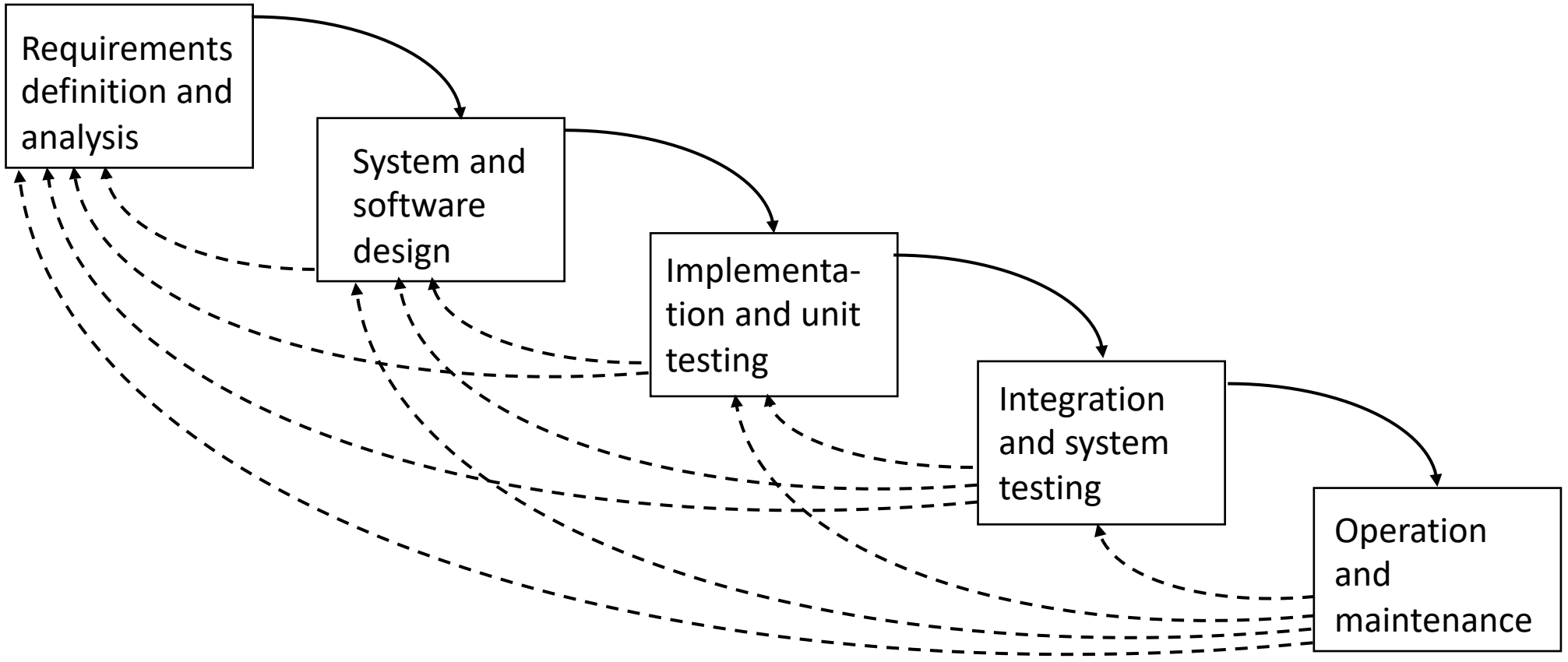
Fielded Product Life

- Routine maintenance, patching
  - Responsibility of engineering in small organizations
  - Responsibility may be in different group than one that manufactures product
  - Example of failure: 2017 Equifax breach believed due to failing to install an important system patch, resulting in breach of financial information for hundreds of millions of people
- Customer service, support organizations
- Retirement or decommission of product

Waterfall Life Cycle Model

- Requirements definition and analysis
  - Functional and non-functional
  - General (for customer), specifications
- System and software design
- Implementation and unit testing
- Integration and system testing
- Operation and maintenance

Relationship of Stages



Agile Software Development

- Software development is creative process, always changing, never really completed
- Leads to agile methodologies
  - Focuses on working together
  - Agile team efficiently works together in their environment
  - Team engages customer as a member of the team, developing requirements and scoping of the project
  - Accept, adapt to rapidly changing requirements
    - Allows for continuous improvement

Agile Methodologies

Term “Agile software development” used to describe several Agile methodologies

- Scrum
- Kanban
- Extreme Programming (XP)
- Others
  - Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM), Pragmatic Programming

In all, evidence of trustworthiness for assurance adduced *after* development

Scrum

- Split project into small parts that can be done in a short timeframe (called a *sprint*)
  - This *product backlog* created by product owner, who represents customer, product stakeholders
- Scrum team agrees on a small subset from top of backlog, decides how to design, implement it
  - Goal: complete this within the sprint
- Every day, team meets to evaluate progress, adjust as needed to get a workable solution within each sprint
  - At the end, work completed should be ready to ship, demo, or put back into backlog if not complete
- Iterate until product complete

Kanban

- Identify lanes of work: to be done, in progress, completed, deployed
- Each lane except the last has limit on how many items can be in that lane
  - Based on staff available to perform the work
- Teams take item off to be done lane, work on it until completed
  - When implemented correctly, team is completing work on top item in lane when another item arrives
- Goal: deliver product to customer within expected timeline
  - Methodology originated at Toyota

Extreme Programming

- Rapid prototyping and “best practices”
- Project driven by business decisions
- Requirements open until project complete
- Programmers work in teams
- Components tested, integrated several times a day
- Objective is to get system into production as quickly as possible, then enhance it

Models

- Exploratory programming
  - Develop working system quickly
  - Used when detailed requirements specification cannot be formulated in advance, and adequacy is goal
  - No requirements or design specification, so low assurance
- Prototyping
  - Objective is to establish system requirements
  - Future iterations (after first) allow assurance techniques

Models

- Formal transformation
  - Create formal specification
  - Translate it into program using correctness-preserving transformations
  - Very conducive to assurance methods
- System assembly from reusable components
  - Depends on whether components are trusted
  - Must assure connections, composition as well
  - Very complex, difficult to assure

Security: Built In or Add On?

- Think of security as you do performance
  - You don’t build a system, then add in performance later
    - Can “tweak” system to improve performance a little
    - Much more effective to change fundamental algorithms, design
- You need to design it in
  - Otherwise, system lacks fundamental and structural concepts for high assurance

Reference Validation Mechanism

- *Reference monitor* is access control concept of an abstract machine that mediates all accesses to objects by subjects
- *Reference validation mechanism* (RVM) is an implementation of the reference monitor concept.
  - Tamperproof
  - Complete (always invoked and can never be bypassed)
  - Simple (small enough to be subject to analysis and testing, the completeness of which can be assured)
    - Last engenders trust by providing evidence of correctness

Examples

- *Security kernel* combines hardware and software to implement reference monitor
- *Trusted computing base* (*TCB*) consists of all protection mechanisms within a system responsible for enforcing security policy
  - Includes hardware and software
  - Generalizes notion of security kernel

Adding On Security

- Key to problem: analysis and testing
- Designing in mechanisms allow assurance at all levels
  - Too many features adds complexity, complicates analysis
- Adding in mechanisms makes assurance hard
  - Gap in abstraction from requirements to design may prevent complete requirements testing
  - May be spread throughout system (analysis hard)
  - Assurance may be limited to test results

Example

- 2 AT&T products with same goal of adding mandatory controls to UNIX system
  - SV/MLS: add MAC to UNIX System V Release 3.2
  - SVR4.1ES: re-architect UNIX system to support MAC

Comparison

- Architecting of System
  - SV/MLS: used existing kernel modular structure; no implementation of least privilege
  - SVR4.1ES: restructured kernel to make it highly modular and incorporated least privilege

Comparison

- File Attributes (*inodes*)
  - SV/MLS added separate table for MAC labels, DAC permissions
    - UNIX inodes have no space for labels; pointer to table added
    - Problem: 2 accesses needed to check permissions
    - Problem: possible inconsistency when permissions changed
    - Corrupted table causes corrupted permissions
  - SVR4.1ES defined new inode structure
    - Included MAC labels, DAC attributes
    - Only 1 access needed to check permissions

Computers and Elections

- This looks at the technology
  - Procedures, policies equally important, but require a different type of analysis ("process modeling", used to model software development, can be applied here_
- Does using computers in an election process:
  - Introduce new ways for attackers to compromise the election, or prevent voters from voting?
  - Stop any of the previous ways for attackers to compromise the election, or provide new ways to enable voters to vote?

Some Terms for E-Voting Systems

- BMD: Ballot Marking Device
  - Marks a paper ballot
- DRE: Direct Recording Electronic
  - Stores votes (ballots) electronically
- DRE + VVPAT: DRE + Voter Verified Paper Audit Trail
  - A DRE that also prints a paper record of the votes (ballots) cast on it
- PCOS: Precinct Count Optical Scanners
  - Used to count paper ballots at the precinct (polling station); these are stored electronically and the memory cards used to transfer results to central vote tabulator

Some Terms for Elections

- Race
  - An element on a ballot that people vote on
- Overvote
  - More votes cast by a voter in a particular race than is allowed for a voter
- Undervote
  - Fewer votes cast by a voter in a particular race than is allowed for a voter
- Example
  - Race is 3 open seats for city council, 5 candidates for those seats
  - I vote for 2 of them, not 3: that's an undervote and it counts
  - I vote for 4 of them, not 3: that's an overvote and it doesn't count

How an Election Works in Yolo County, CA

- Voters:
  - Go to polling station, give name
  - Get ballot, enter booth, vote using marker to mark ballot
  - Put ballot in protective sleeve, leave booth
  - Drop ballot into ballot box
    - If provisional or conditional, put ballot and sleeve into envelope with voter's name, reason for the challenge (provisional) or condition (conditional) on the *outside*
- Vote-by-mail voters:
  - Fill in ballot
  - Put ballot into inner envelope
  - Put inner envelope into mailing envelope; sign the *outside* and mail it in

End of the Day

- Election officials take ballot box to County seat
- Election officials remove ballots from envelopes
  - Provisional and conditional ballots handled separately
- Ballots counted, put into bags marked with precinct and count
- Ballots removed from bag, run through automatic counters
  - Humans intervene when problems arise
  - Intermediate tallies written onto flash cards
  - Every so often, cards removed, walked to tally computer, inserted, votes counted
- Reported tallies periodically updated, given for posting to web

And Then . . .

- All places have provisional ballots
  - These are cast when it is unclear if the person is allowed to vote
  - In California, ***always*** on paper, never electronic
- California allows conditional ballots
  - These are cast by folks who register at the election (same day registration)
- Conditional and provisional ballots must be validated before being counted
- California also allows mail-in ballots arriving up to 3 days after Election Day to be counted

The Canvass

Required by California law:

- Ballots for 1% of precincts counted by hand
  - Chosen with throw of dice; if some races not in precincts selected, add more in until all covered
  - Some counties have legal authority to use risk-limiting audit as well or instead
  - In California, you *must* use paper for this (hence, all DREs have VVPATs)
- Compared to tallies from election
  - If different, must be reconciled
- Certify final counts to Secretary of State
  - Has to be done within some number of days after election

Some Election Requirements

- Voter validation (authenticated, registered, has not yet voted)
- Ballot validation (voter uses right ballot, results of marking capture intent of voter as required by law)
- Voter privacy, secrecy (no association between voter, ballot; includes preventing voter showing others how he/she voted)
- Integrity (ballots unchanged, votes tallied accurately)

Some Election Requirements

- Voting availability (voter must be able to vote, materials must be available)
- Voting reliability (voting mechanisms must work, even under adverse circumstances)
- Election manageability (process must be usable by those involved, including poll workers)
- Election transparency (audit election process, verify everything done right)

What Should an E-Voting System Do?

- Replace manual activity, existing technology used in election process with better technology
  - Better in the sense of improving some aspect of the election process
- Examples
  - Easier to program ballots than print them
  - Can handle multiple languages easily
  - Easier to tally than hand counting

Assurance

- Provide sufficient evidence of assurance to target audience that using e-voting systems makes elections at least as secure, accurate, etc. as elections without them (that is, using paper ballots)
- Who is “target audience”?
  - Computer scientists, election officials, politicians, *average person*

A Smattering of Problems

- Boone County, IN, 2003: 144,000 votes cast in a county with about 6,000 voters
- In 2006, polls opened late in several California (CA) counties (San Diego, Alameda, Plumas, Kern, Solano) due to system problems
- South Bronx, NY, 2010: a scanner miscounted 69/103 (70%) of ballots in Sep., then 156/289 (54%) in Nov.
- Los Angeles, CA, 2020: electronic poll books had connectivity problems, resulting in unacceptably long lines; BMDs failed, had paper jams

Results of Testing

- 2003: Johns Hopkins people analyzed voting system program
- 2004: RABA rigged mock election in about 30 minutes
- 2006: Florida CD-13 election *post mortem*
- 2007: California Top-to-Bottom Review, Ohio EVEREST review
- 2011: Washington DC internet voting test compromised
  - And the friendly attackers threw out the hostile ones
- 2014: Analysis of Estonia e-voting systems: many vulnerabilities found
- 2020: Voatz mobile voting app based on “blockchain technology”: many vulnerabilities found

How to Get Better

- You need both standards and testing
- They must be independent of the developers of the systems
- They need to consider the users, operators, and maintainers of the systems
- Reports should show what tested, why, and how
- For e-voting systems, penetration testing is a *must*

Add in the Internet

- It will enable authorized voters who cannot vote due to distance (or other factors) to do so
- It will increase authorized voter participation
- It will bring our elections into the modern, technological world
- It will be cheaper because we don't have to store the paper ballots

Problem:

- Election systems are now accessible to many more people than authorized voters!

Where Would Attackers Strike?

- Probably not regular, individual electronic voting systems
- But attack the vendors and change the programs that run on those systems, or on the tallying systems
- Or hit the voter registration databases to disenfranchise voters

Remote Voter Verification of Ballots

- Trick here is to protect against the validating mechanism being corrupted
- Example: we examined a system that enabled voters to check that their ballots were recorded correctly, and counted correctly, remotely
  - Used very neat cryptography, done by experts
  - We simply changed the web page on which the information that the user used to do the validation – no cryptography involved!

Moral: attackers don't have to rig or corrupt an election

They just have to make you *think* they did!

Blockchains

- Background
  - Take ballot or chain of ballots and compute a hash from them
  - Encrypt this with a cryptographic key you keep secret (private key)
  - Publish the inverse cryptographic key (public key) so others can verify the small value was not changed
- For voting: many proposals for handling the chains

Why Blockchains Fail for Elections

- Problem #1: denial of service (already discussed)
- Problem #2: how are those cryptographic keys generated?
  - A. Voter generates the pair (this is how it's usually done for other uses), and publishes the public key
  - A′. I vote multiple times, possibly under the name of a different voter each time. Prove I was the one who did this, and determine which votes are mine.
  - A″. I want to sell my vote. I give my private key to the purchaser. She can use the public key to verify that is my private key, and then see how I voted by finding the specific ballot added using that public key.
  - B. Election officials assign key. Now *they* can determine how I voted!

How Not to Test Voting Over the Internet

- Occasional bills in various legislatures to do a “pilot study” using Internet voting in a real election
- A valid test requires knowing “ground truth”, that is, what the results of the election should be
- How do you know this in a real election?