Introduction to Medical Software

Introduction to Medical Software on Coursera (Yale University)

Week 1 Intro to Medical Software and Digital Health

- Overview
  - A guided tour of medical software
    - Software qualifies as a medical device when it is used to treat/diagnose an individual patient
    - Funding in digital health applications is rapidly increasing
    - Validation: does it conform to use cases (user needs)?
    - Verification: does it conform to system requirements?
- The Regulatory Landscape
  - ISO 13485: Quality Management Systems
  - ISO 14971: Risk Management
  - IEC 62304: Software Life Cycle Process
  - IEC 62366: Usability Engineering
- The Digital Transformation of Healthcare
  - As a share of GDP, the US's health spending accounted for 17.7% in 2019 ($3.8 trillion or $11,582 per person)
  - **Centers for Medicare and Medicaid Services**
  - **US Healthcare from a Global Perspective, 2019: Higher Spending, Worse Outcomes?**

Week 2 Medical Software Regulation

- Medical Software Regulation (FDA and IMDRF)
  - FDA is responsible for protecting public health by ensuring safety, efficacy, and security of medical devices.
  - https://www.fda.gov/medical-devices/digital-health-center-excellence/software-medical-device-samd
- FDA History and Original Regulations
  - Food and Drug Act 1906 – poisonous food additives
  - Food Drug and Cosmetics Act 1938
- Contemporary Guidance and Regulatory Processes
  - Validation: Final device correctly and consistent meets user needs

- Verification: Specified requirements (design inputs) are fulfilled by design outputs
- Design Input: Per IEC 62304, interpretation of customer needs into documented medical device requirements
- Specification: Per 62304, documentation of what software does to meet customer needs and design inputs
- Design Output: results of design effort at each design phase and at the end of the total design effort
- Design Review: systematic examination to evaluate capability of design to meet requirements
- Design History File: compilation of records describing design history of the finished device
- EU Medical Device Regulation 2017/745 (MDR)
    - Scope and Definitions
    - Placing into market and other issues
    - Traceability and registration
    - Notified bodies
    - Classification and Conformity Assessment
    - Clinical Evaluation
    - Post Market Surveillance, Vigilance, Market Surveillance
    - Member States Cooperation
    - Confidentiality, Data Protection
    - General safety and performance requirements
    - Technical documentation
- Reading: **Promoting Safe & Effective Drugs for 100 Years**
- Reading: **The Dalkon Shield Disaster**
- General Principles of Software Validation
    - Planning, verification, testing, traceability, configuration management
    - Typical tasks supporting validation
        - Quality planning
        - Requirements
        - Design
        - Construction or coding
        - Testing by developer
        - User site testing
        - Maintenance and changes

        - See *General Principles of Software Validation; Final Guidance for Industry and FDA Staff*—covers scope, principles, activities, testing, and resources
    - Validation and verification activities should be conducted throughout the entire life cycle
    - Software updates can be trouble; FDA's analysis of 3140 medical device recalls conducted between 1992 and 1998 reveals that 79% were caused by software defects introduced *after* initial production and distribution, i.e. software updates
    - Because software cannot be tested forever, software validation is a matter of developing a "level of confidence" that the device meets all requirements with regard to safety risks
    - Small changes in code can create unexpected and significant problems
    - Personnel who make maintenance changes may not have been involved in original development
    - Software requires greater level of scrutiny than hardware
- The IMDRF SaMD Guidance Documents
    - International Medical Device Regulators Forum
    - Software as a Medical Device
    - Goal is to assess and minimize risk; risk is categorized by probability and severity
    - Medical software has a long lifetime
        - Select operating system/dependencies such that they will be around
        - Do not use bleeding edge libraries and systems that have not stood the test of time
        - Code for those who will come after
- The Regulatory Process
    - Efficacy
    - Safety -> Usability
    - Security -> Cybersecurity
    - Software is mostly Class II
    - Review processes
        - 510(k) – FDA determines that device is substantially equivalent to predicate, previously commercialized product

- De novo – novel devices with Type I or Type II classification based on reasonable assurance of safety and effectiveness for the intended use in the absence of predicates
- Premarket Approval (PMA) – demonstrate reasonable assurance of safety and effectiveness; this is complicated and expensive
- MDR -> classes I, IIA, IIB, and III; default class for software is IIa
    - If it can cause death or irreversible deterioration of person's state of health -> class III
    - Review is done by Notified Bodies, private companies authorized by government competent authorities
    - Results in conformance certificate CE Mark that is placed on device
- Process in China: NMPA, formerly CFDA
    - Is a unique process but shared some similarities with US
    - Testing needs to be to Chinese National and Industry Standards

- Emerging Issues: AI/ML
    - Good review in LeCun, Bengio and Hinton's seminal Nature Paper (May 2015)
    - Huge promise as solution to many problems, but hype must be tempered with reason (more in Week 11)
    - Major challenge is collection of large amounts of training data
    - Data needs to be "cleaned" and "annotated"
    - FDA Draft Guidance (2019)
    - Regulators are working on approach to software that updates itself
    - Can you create a solution that mitigates issues on its own?
    - General trend: proper design, quality management procedures provide assurance good results were not obtained by chance

Week 3 The Healthcare Environment

- The Healthcare Environment (EHR, PACS, Data Privacy, and Cybersecurity)
- The Clinical Environment

        - Patients, Doctors, and ???
        - Clinical Information Technology
            - Electronic Health Records (EHR)
                - Notes from patient visits
                - Most common protocol is HL7, e.g. EPIC
                - Reading: **Development of the EHR**
            - Electronic Medical Records (EMR)
            - Picture Archive and Communication System (PACS) – Imaging Database
                - Imaging modalities:
                    - 1895: X-rays
                    - 1950s: Nuclear medicine (SPECT, PET – radioactive glucose to measure metabolic activity in cancer)
                    - 1970s: Ultrasound (low energy, cheap, portable)
                    - 1970s: X-ray CT (first true 3D images, computers meet imaging)
                    - 1980s: MRI
            - Demo of EHR and PACS
- Cybersecurity and Data Privacy
            - Data Privacy
                - HIPAA – Health Insurance Portability and Accountability Act 1996
                - PHI: Protected Health Information
                - ePHI: Electronic Protected Health Information
                - HIPAA implications for software design: do not store anything other than bare minimum of patient data and store in such a way that it does not constitute ePHI; anonymization
                - DICOM: Digital Imaging and Communications in Medicine — the international standard for medical images and related information
            - Cybersecurity
                - Definition: a state where information and systems are protected from unauthorized activities, such as access, use, disclosure, disruption, modification, or destruction to a degree that the related risks to confidentiality,

integrity, and availability are maintained at an acceptable level throughout the life cycle.
- FDA has draft guidance, categorizes into Tier 1 – higher risk, multiple patients potentially impacted, and Tier 2 – standard risk.
- Designing a trustworthy device: identify, protect, detect, respond, recover
    - Prevent unauthorized use
    - Authenticate users
    - Encrypt communications
- Cybersecurity cannot be achieved by a single stakeholder; it requires a concerted effort of diverse stakeholders (government, manufacturers, healthcare institutions, users)
- Reading: **Framework for Improving Critical Infrastructure Cybersecurity**

Week 4 Quality and Risk Management

- Quality and Risk Management
    - System and set of policies that govern how a company operates with the objective of meeting regulations
    - ISO 9001
    - ISO/IEC/IEEE 90003
    - ISO 13854
    - IMDRF (International Medical Device Regulators Forum) Software as a Medical Device (SaMD) Application of Quality Management System
- Quality Management Systems
    - Organizational structure for implementing quality management
- Introduction to Risk Management
    - Requires balanced evaluation of safety and security
    - Document and record control
    - Measurement analysis and continuous improvement
    - Patient safety and clinical environment considerations (medical)
    - Technology and systems environment considerations (software)
    - Unless a product is designed and produced in a proper organization, it is unlikely to be of high-quality

- IEC 62304: There is no known method to guarantee 100% safety for any kind software
- When an engineer walks into a new situation, the first questions are
    - What could possibly go wrong here?
    - What can I do to either make sure that it does not?
    - What can I do so if something goes wrong, that it does not cause too much damage?
    - Similar concept to defensive driving
- ISO 14971: General requirements for risk management systems
    - Hazards -> Sequence of events -> Hazardous situation -> Harm
    - Not all hazardous situations result in harm
    - Risk analysis: systematic use of available info to identify hazards to estimate risk
    - Risk control: process in which decisions are made and measure implemented to reduce risks or maintain within specified levels
    - Residual risk: risk remaining after control measures have been implemented
    - Risk management: systematic application of management policies, procedures, and practives to the tasks of analyzing, evaluating, controlling and monitoring risk
  - Categories of hazard
    - User – especially if used at home
    - Application – does it restrict availability
    - Device – e.g. screens too small
    - Environment – e.g. connectivity
    - Security (cybersecurity)
    - Cost risk
    - Technical risk
    - Schedule risk
    - Organization risk
    - Market risk
    - Regulatory risk
  - The Risk Management Process
    - Estimation of risk
      - Probability vs severity

            - Hazard -> Sequence of Events -> Hazardous Situation -> Harm
        - For each hazardous situation do:
            - Evaluate the estimated risks
            - Determine if risk is acceptable or not
            - If acceptable, document and stop
            - If not, perform risk control activities and reduce the risk
    - Risk Control Methods
        - Change design to eliminate possibility of hazard occurring
        - Add preventative measures in software to reduce probability of harm
        - Add safety information measures
    -

Week 5 Software Development

- Software Development Life Cycle
    - IEC 62304 provides framework for medical device software lifecycle for safe design and maintenance
    - Risk classification
        - Class A – low
        - Class B – unacceptable risk but injury not serious
        - Class C – unacceptable risk that can lead to serious injury/death
    - What is a life cycle?
        - System requirements
        - Software Design
        - Software Implementation
        - Verification and Validation
        - Release and Maintenance
    - Different types of lifecycle:
        - Waterfall: one direction, one shot from needs to delivery
        - Incremental: needs/requirements done once, then multiple development cycles each adding capabilities
        - Evolutionary: user needs cannot be fully understood so requirements are not fully defined up front; interim software distributed on an iterative basis
- Life Cycles and the IEC 62304 Standard

        - Maintenance is often higher risk because the original developers may no longer be the ones making changes, and starts with bugs rather than requirements, moreover software exist in maintenance part of lifecycle longer than it lives in development; 80% of FDA issues are found in version updates
        - Design controls discipline: must operate “under control”; if problem is found during design, stop, fix, then resume. We don’t want to make “on the fly” decision without updating the design.
        - Software Process Models: https://www.thomasalspaugh.org/pub/fnd/softwareProcess.html
- The Example Project and a Guided Tour of the Process
    - Image-guided Therapy
        - Planning
            - Acquire pre-procedure images
            - Analyze + align images
        - Initialize
            - Register image to patient
        - Guidance
            - Real-time tool tracking, imaging, account for motion
            - Provide Feedback
    - Brainlab vector-vision cranial
    - Image-guided prostate biopsy
    - https://www.scientificamerican.com/issue/sa/1999/06-01/
    - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2796181/
    - Step 1: User needs
        - What are they trying to do right now?
        - How are they doing it?
        - What would they like to be able to do?
        - How will the product help them do it?
        - May need to build prototypes to better identify issue
    - Step 2: System Requirements Specification
        - What are those “things” that the system shall do?
        - Captured in Systems Requirements Specification
        - Usage scenario can be summarized as user, use, scenario
        - Functional requirements: what system must do to satisfy needs
        - Non-functional requirements: capabilities must do well to ensure safe, secure, pleasing user experience
        - Some user needs map to more than one requirement and some requirements satisfy more than one user need

        - Use simple and direct writing; use words consistently
        - System specifications act as bridge between management and senior software designers. Software design acts as a bridge between senior software designer and junior engineers
    - Step 3: Construction and Verification
    - Step 4: Validation
    - Step 5: Deployment
    - Step 6: Maintenance
    - Step 7: Retirement
    - Ensuring Information Transmission
        - User -> Business Analyst (user needs) -> Senior design team (system requirements) -> Senior software engineers (software design) -> Junior developers (coding)
        - Explaining the “why” is very important down the chain
        - Don’t want to overload junior engineers with information
    - Sufficient statistic: summary of dataset that gives all the information you need for a task; the appropriate summary that users need to do their job, e.g. how many pizza’s should I order?
    - Traceability
        - Systems requirements to risk analysis
        - Systems requirements to software design
        - Source code as implemented back to software design
        - Source code as implemented forward to appropriate tests
        - Testing

Week 6 User Needs and System Requirements

- User Needs & System Requirements
    - How to demonstrate safety, efficacy, and security?
    - What? (Abstract) and How? (Concrete)
    - What now, how now, what future, how future
    - Learn about clinical problem and the patient
- Identifying User Needs
    - Initial meeting
    - Establish good working relationship
    - Make a good first impression, dress appropriately
    - Figure out roles, responsibilities, and interests of each person
    - Do not rush to conclusions
    - To somebody with a hammer, everything is a nail

        - Known unknowns (easy), unknown unknowns (hard)
        - Ask questions like:
            - Is there anything else I need to know?
            - Would you like to add anything?
        - Listen! Keep ears open. It is better to ask more questions than to pretend you understand what is going on and pay the price later
        - Try to restate the problem in your own words
        - Context of the problem
            - Understand why the problem presented is significant
            - What has been tried before to address this?
            - Is the primary driver quality, affordability, or user skill?
        - Identify the lay of the land
            - What comes before this and what comes after?
            - What happens to the patient prior to this point and what will happen after this?
            - Where is the data coming from?
            - Where does my output need to go?
            - What constraints does this impose on the problem?
            - Is this a time limited process? Are there things that we can do ahead of time?
            - Ask about safety/privacy issues and other limitations in terms of what computers can be used and what cannot
            - Ask about regulatory/safety issues that may govern aspects of the problem
            - Ask about domain specific conventions
            - Do not let clinician's idea of the solution necessarily bias the solution
            - Current way of doing things may not be due to "technical incompetence"
                - There may be very good reasons for current practices
                - Try to identify what is critical and what is accidental
                - Current methods may also be constrained by legal/regulatory issues
            - Most needs may step from how clinicians interact with software
            - Consider all stakeholders
- Systems Requirement Specification (SRS)
    - Bridge between user needs and actual design
    - Things to consider in software development
        - use environment

        - data integrity
        - underlying platform/OS
        - Non-functional aspects e.g. network, hardware performance requirements
        - Stakeholders
    - Template:
        - Cover page
        - Intro
            - Goals and Objectives
            - Statement of Scope
            - Definitions
            - References
            - Significance
        - Overall Description
            - Intended Users
            - Intended Uses
            - Intended Use Environment
                - Physical
                - Standards
                - Legacy Environment
                - Platform
            - Functional Requirements
            - Non-functional Requirements
    - “Shall” indicates not optional
    - https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6774318

Week 7 Design and Usability

- Software Architecture Design and Usability Engineering
- Software Design
- Usability Engineering
    - IEC 62366
    - Human factors considerations, inputs of device usage:
        - Use Environment
        - User
        - Device User Interface
        - Outcomes of device use:
            - Correct use: safe and effective use
            - Use error: unsafe or ineffective use

        - Usability improves safety
        - Two types of evaluation: formative and summative
        - FDA Guidance on Applying HFE to Medical Devices: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/applying-human-factors-and-usability-engineering-medical-devices
- Digital Behavioral Health
    - https://www.aha.org/system/files/media/file/2019/06/Market_Insights_Be_havioral_Health_Landscape.pdf
    - https://www.fiercehealthcare.com/tech/funding-for-digital-behavioral-health-startups-surged-amid-covid-19-pandemic

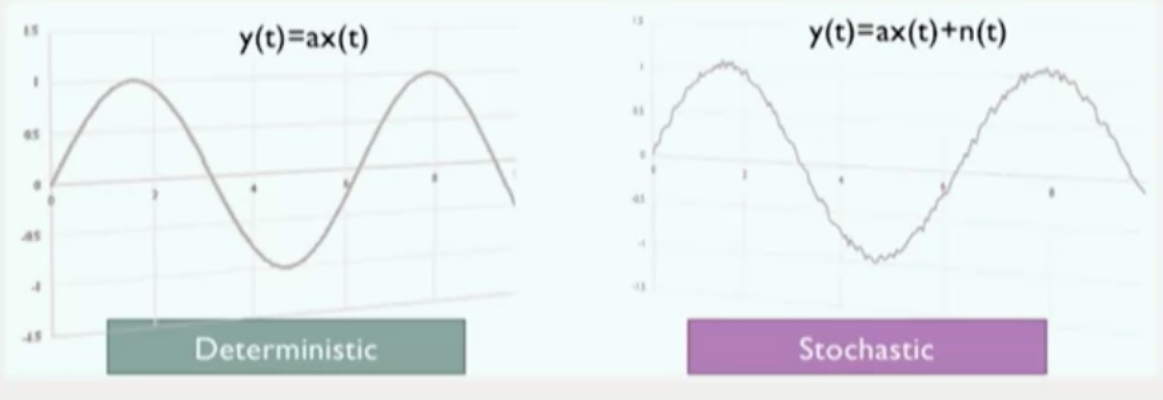
Week 8 Construction and Testing

- Construction and Testing
    - Implementation -> Low Level Testing -> Verification
- Software Construction (or Coding)
    - Follow standard best practices taught in most computer science courses
    - Code should be modular, well documented, easy to read
    - Use defensive programming techniques
    - Use shared repository and appropriate branches
    - Do not use latest and greatest features or unnecessary complexity
    - Be mindful of internationalization
    - Medical software packages are maintained for extended periods of time, e.g. 10, 15 years
    - Code is highly likely to be reviewed and modified by programmers that were not present at the time of the design
    - Optimize for readability instead of compactness
    - Use revision control systems, e.g. Github
        - Git clone: point to an existing repo and make a clone or copy of that repo in a new directory at another location
        - Git commit: snapshot of project's currently staged changes
        - Git push: upload local repo content to remote repository
- Software Testing
    - Automated regression testing
    - Interactive user testing
    - If we find no errors, all we know is that testers found no errors
    - White box vs black box testing

        - White box: detailed knowledge of code, automated regression, goal is exercise units over a range of inputs to ensure correct behavior
        - Black box: no knowledge of code
- Reduce complexity to allow testing
- Code with the testing in mind
- What is software testing?
    - Specified conditions
    - Expected results
    - Actual results
    - Assessment
- Levels of testing
    - Unit
        - Testing of individual module of code
    - Integration
        - How information is transferred module to module
    - System
        - Testing of system as a whole
        - Smoke tests: simply try to do something with the software and see if it works. Huge benefit is that these tests involve the actual software, not small programs.
- Be mindful that test code is also code! And should be reviewed
- Use scripts, in the theater sense. Plan out sequence of tests and expected results.
- Integration testing in realtime fMRI
    - Created system to simulate inputs for purpose of testing
    - Simulation can catch most errors prior to doing final set of tests in real world
- Verification testing
    - IEC 62304 Risk Classification
    - Testing code needs to be reviewed in the same way as actual product code; testing code can have bugs
    - Adding randomness may help reveal issues
- Testing plan
    - Consider inexperienced testers, difficult conditions, and ill-defined tasks
    - Regression testing for automated parts
    - Test for desirability!
    - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3644525/

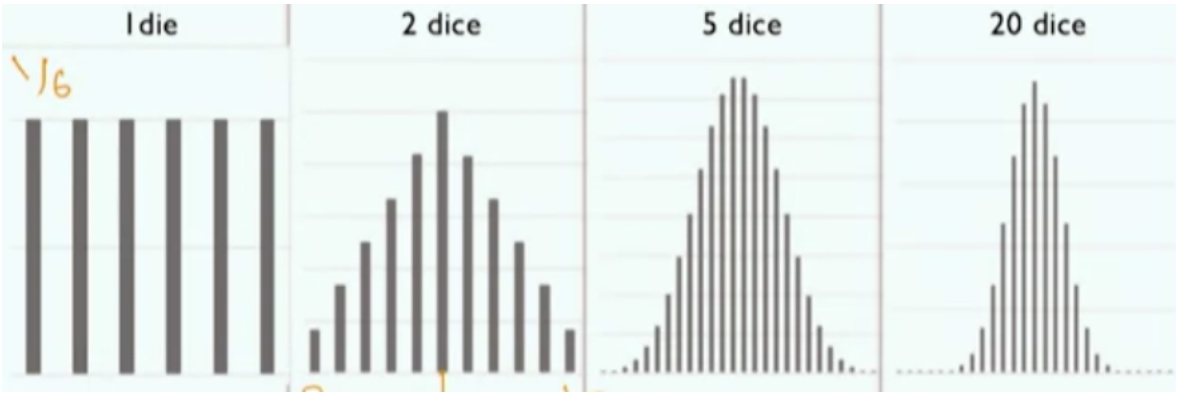
Week 9 Probability and Statistics

- Probability Overview
    - Intro to Probability
        - Deterministic vs stochastic
        - Deterministic: can compute exact output given inputs
            - x(t) is the input, y(t) is the output, parameter = a
        - Stochastic system: can “estimate” the output with some level of uncertainty
            - n(t) is random noise



        - Probability density function; coin flip has binomial probability density—a discrete PDF (probability density function)
        - Continuous probability density function; e.g. normal distribution
    - Multiple Events
        - Bayes’ Rule: probably of a given
        - P(a|b) = (p(b|a)p(a))/(p(b)
            - P(a|b) = “Probability of a given b”
            - Belief updating
- Statistics Overview
    - Statistic: measure extracted from a dataset
    - Choose appropriate statistic:
        - For ordering pizza, ask for average slices per person
        - For building codes, ask what maximum resistance needs to be
    - Two types of estimation:
        - Point estimation: single number, best guess at a population parameter

        - Interval estimation: two numbers, lower and upper bounds for specific parameter, associated with confidence level e.g. 95% that parameter is within the bounds
        - Central Limit Theorem: under certain conditions, the sum of lots of random variables has a normal distribution
        - Law of Large Numbers: under certain conditions, the estimates of parameters (e.g. mean) converges to the true value



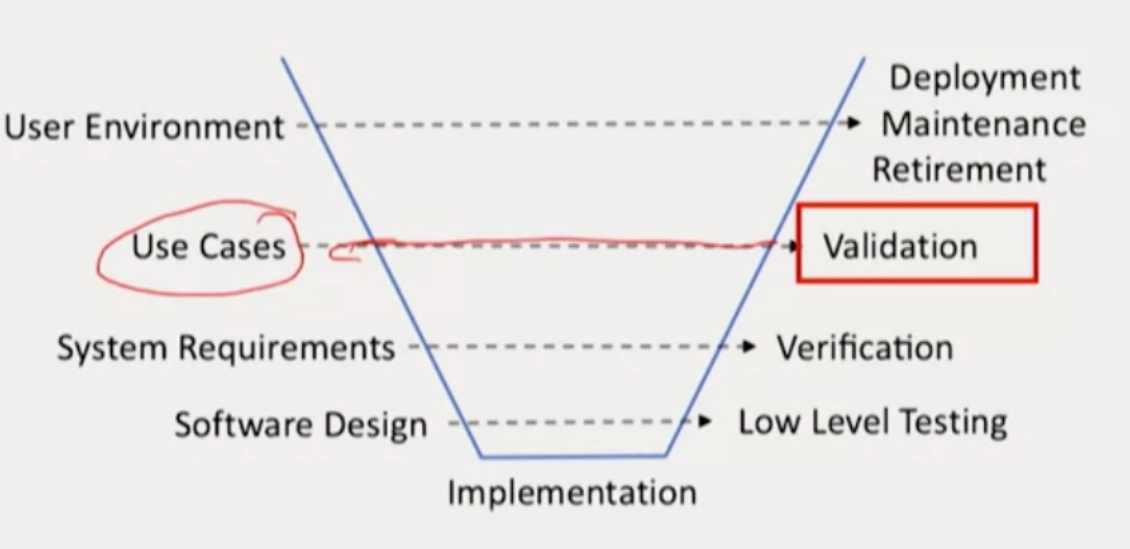
- Estimating probability functions
    - Acquire measurements from large enough sample of representative situations
    - Use these measurements to construct probability density function using either parametric or non-parametric methods
        - Parametric: assume pdf has known shape with unknown parameters
            - Is model appropriate?
            - Assume a Model Distribution (e.g. Gaussian)
            - Computer mean and standard deviation from the sample
            - Plug into our model
        - Non-parametric:
            - Used when we do not have a good parametric model
            - Advantage: weak assumptions about the form of the distributions
            - Disadvantage: need more data as we are estimating the whole shape
            - Imagine curve with one random spike representing outliers
- Signal Detection
    - Is there evidence of cancer in this mammogram?
        - Typically boils down to setting a threshold f(x)>t

        - **Sensitivity**: we do not miss any cancers; ratio of correct positive diagnoses compared to total actually sick patients
        - **Specificity:** when we give a cancer diagnosis, it is true
        - Easy to maximize each separately, but want to minimize false negatives and positives, and ensure positive calls are accurate
        - How to set "t" threshold
            - What happens to performance as we vary t?
            - Receiver operator characteristic curves (ROC)
        - https://www.sciencedirect.com/science/article/abs/pii/S1546144009002762?via%3Dihub
- Tools for Clinical Trials
    - Real significance vs statistical significance
        - We want to know if drug reduces length of hospitalization
            - Recruit population
            - Randomized trial
            - Effect = mean reduction of hospitalization time between groups
        - If effect is small, we are done, no real-world significance even though there may be statistical significance
    - How does significance testing work?
        - Perform a trial where you have two groups where only difference is drug vs placebo
        - Measure outcome statistics i.e. difference in hospital time (D)
        - Assume Null Hypothesis that there is no effect
        - Compute probability of getting (D) at least as big as what you have measured if Null is Correct
        - If this is small, reject the Null
        - Essentially, we measure p(D>=d | H naught) and check if this is smaller than a threshold (typically 0.05 or 0.01).
        - **"We were able to reject the null hypothesis that the drug had no effect as the hypothesis testing resulted in a p<0.05"**
            - **This means there was a less than a 5% chance of observing the data we did if the drug had no effect**.
        - Where does the p<0.05 come from? R.A. Fisher *Statistical Methods for Research Workers*
        - Unpaired T-Test
            - We have two groups size n1 and n2
            - Test hypothesis that the two groups are different

        - Assuming each group has Gaussian distribution and similar variances
        - Compute means and st. devs of each group to compute t-statistic, distribute with student's t-distribution with degrees of freedom = v
        - **Compute survival function p(t>T)**
        - If small, reject null
    - Power analysis
        - Alpha (e.g. 0.05) -> significance level p<a
        - Type I error: reject null when no effect (false positive)
        - Type II error -> not reject null where there is effect (false negative)
        - Statistical power: probability that a test will correctly reject a false null hypothesis
        - High power means we are less like to have a false negative
    - **We compute probability of seeing this data if the null hypothesis is true, we do not computer the probability that the null hypothesis is false**
- Importance of randomization
    - Clinical trials: structured experiments where we try to demonstrate that a proposed intervention has a significant positive effect in a certain population that is not observed simply by chance
    - Randomized: participants are assigned to the treatment or control group by chance
    - Allows us to eliminate hidden biases of selection

Week 10 Software Validation

- Software Validation, Deployment, Maintenance and Retirement
    - Validation: ensuring that software meets needs of user, not that it necessarily works correctly (verification); necessary but not sufficient



- Efficacy, safety, and security
  - Safety -> Usability (Week 7)
  - Security -> Cybersecurity (Week 3)
- Valid Clinical Association
  - Is there valid clinical associate between your SaMD output and the targeted clinical conditions?
- Analytical Validation
  - Does your SaMD correctly process input data to generate accurate, reliable, and precise output data?
- Clinical Validation
  - Does use of your SaMD's accurate, reliable, and precise output data achieve your intended purpose in your target population in the context of clinical care?
- https://www.fda.gov/medical-devices/digital-health-center-excellence/software-medical-device-samd
- Introduction to clinical trials
  - Clinicaltrials.gov
  - Human subjects are prospectively assigned to one or more interventions
  - Analysis plan is fully defined prior to data collection
  - Randomized
- Two different types of IDEs (Investigational Device Exemption)
  - **Feasibility Study**
    - Small study to establish safety of device and potential effectiveness

        - Should follow pre-clinical studies (animal studies and phantom studies)
    - **Pivotal Study**
        - Larger study, serves as primary clinical support for full FDA application
        - Demonstrates reasonable assurance of safety and effectiveness
        - Endpoints and number of subjects are driven by prior statistical analysis
        - FDA review is more involved
    - Process is similar for drug studies
        - Phase I – safety testing in small group
        - Phase II – larger group, determine safety and effectiveness
        - Phase III – large, multi-site, last step prior to requesting FDA approval
        - Post-market trials performed to evaluate long-term safety
    - Helsinki Declaration 1964: research can never take precedence over rights and interests of individual research subjects
    - https://www.wma.net/wp-content/uploads/2016/11/DoH-Oct2013-JAMA.pdf
    - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2912019/
- Creating a validation plan
    - Key questions:
        - Is there a current solution?
        - How does one measure “better” performance?
        - Is this a “good enough” new solution?
    - Define user environment
    - Structure:
        - Goal (hypothesis)
        - Rationale
        - Experimental procedure
        - Statistical Data Analysis
- Deployment, Maintenance, and Retirement
    - Think about deployment at the beginning of the project
    - Steps in deployment
        - Repeatability of delivery, installation, setup, maintenance
        - Risk management
        - Post-market follow-up testing
    - Maintenance and retirement
        - Establish plan for documenting and addressing issues

        - Decommission in a controlled and managed fashion

Week 11 Artificial Intelligence and Machine Learning

- Artificial Intelligence, Machine Learning, and Deep Learning
    - ML is a subset of AI
    - Techniques used to design and train software algorithms to learn from and act on data
    - Pattern classification -> Machine Learning
- Regulatory Issues in using AI/ML for Medical Applications
    - Need to separate training and training set
    - P-hacking -> bad practice in research where multiple analyses are performed to find something statistically significant
        - Neglects to report failed results
- Deep Learning:
    - https://www.nature.com/articles/nature14539
    - https://www.deeplearningbook.org/
    - https://www.nature.com/articles/s41591-018-0316-z
- Regulatory guidance on AI/ML:
    - How to regulate technology that use promise but not fully understood
    - AI Life Cycle DIN SPEC 92000-1:2019 (Germany)
    - China NMPA Guidance
    - FDA Draft Guidance
    - Singapore Guidance
    - Key is establishing what data is needed for testing and validation
    - http://arxiv.org/abs/1907.07374
    - https://www.exhibit.xavier.edu/health_services_administration_faculty/21/
    - https://www.hsa.gov.sg/docs/default-source/announcements/regulatory-updates/regulatory-guidelines-for-software-medical-devices--a-lifecycle-approach.pdf
    - https://arxiv.org/abs/1312.6199
    - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7457546/

Week 12 Business Management Topics

- Business and Management Issues
- Digital Medicine and Business
- Launching a Healthcare Venture

    - https://www.mobihealthnews.com/news/investors-close-out-2020-strong-totaling-138b-digital-health-startups-across-372-raises
    - https://www.forbes.com/sites/greglicholai/2020/01/14/digital-healthcare-growth-drivers-in-2020/
    - https://www.himss.org/resources/four-areas-digital-health-are-driving-future-healthcare
    - This is a description from the FDA's Digital Center of Excellence.
    - https://www.fda.gov/medical-devices/digital-health-center-excellence/guidances-digital-health-content
    - https://www.digital.health/digital-health-venture-funding
    - https://rockhealth.com/insights/h1-2021-digital-health-funding-another-blockbuster-year-in-six-months/

Week 13 Case Studies

- Case Studies
- Therac-25
    - http://sunnyday.mit.edu/papers/therac.pdf
- Fatal Dose AECL
    - http://www.ccnr.org/fatal_dose.html
- Healthcare.gov
    - https://www.nbcnews.com/video/healthcare-gov-builders-say-testing-phase-was-too-short-56243267984
- Health and Human Services Investigation
    - https://oig.hhs.gov/oei/reports/oei-06-14-00350.asp
- Iowa Caucus
    - https://www.youtube.com/watch?v=17AkHXbRJyE
    - https://www.wsj.com/articles/iowa-caucus-results-delayed-by-apparent-app-issue-11580801699
    - https://www.vice.com/en/article/y3m33x/heres-the-shadow-inc-app-that-failed-in-iowa-last-night
- WannaCry
    - https://www.youtube.com/watch?v=5v5gtycGTps
    - https://www.nao.org.uk/wp-content/uploads/2017/10/Investigation-WannaCry-cyber-attack-and-the-NHS.pdf

Week 14 Expert Interviews

1. **Introduction to Organizational Theory for Engineers**: Alka Menon, assistant professor of Sociology at Yale. For more information about her work see webpage https://alkamenon.com/
2. **Understanding Users and Requirements**: Saradwata Sarkar, medical software designer and developer with over 10 years of experience developing software in the medical device industry.
3. **Medical Software Regulation in China**: Sara Li RAC, MS, Senior Regulatory Affairs Consultant at the consulting company China Med Device (http://chinameddevice.com).
4. **FDA Regulation and Process**: Michael Nilo, president and principal consultant at Nilo Medical Consulting. https://www.nilomedicalconsulting.com/
5. **An Introduction to the European Regulatory Process**: Dr. Andrea Biasiucci, competence cluster lead & senior consultant Confinis AG, Switzerland -- https://www.confinis.com/
6. **Cybersecurity and Medical Devices**: Tom Renner, quality, efficiency, and regulatory affairs consultant at Vision28, Redmond, Oregon. (http://www.vision28.com)
7. **Setting Up and Using a Quality Management System**: Dr. Phan Luu, chief science officer of the Brain Electrophysiology Company (BEL)
8. **QMS, ISO 13485 Certification and Risk Management**: Rich Wynkoop, president & CEO of Vision28, Redmond, Oregon. (http://www.vision28.com)
9. **Creating Video Game-based Digital Health Interventions in Youth**: Dr. Lynn Fiellin, founding director of the play2PREVENT lab at Yale (https://www.play2prevent.org). She is a professor of Medicine (General Medicine) Yale Child Study Center, and Public Health at the Yale University School of Medicine. She is also Chief, Fitkin Firm, Yale-New Haven Hospital. Also joining us are Bryce Bjork and Ting Gao, who graduated from Yale College in 2020 and currently spearhead the work of the Chase Bjork Foundation to create Brain Health education programs.
10. **Software Engineering and Medical Software**: Dr. Steve Pieper, CEO of Isomics Inc. He is affiliated with the Surgical Planning Laboratory at BWH, Harvard Medical School, in which role he serves as the lead maintainer of 3D Slicer.
11. **Medical Software Design, Implementation and Testing**: Dr. Rajesh Venkataraman, director of research and development at Eigen Health Services LLC. http://www.eigen.com
12. **Open Source, Open Science and Medical Software**: Dr. Andinet Enquobahrie, Ph.D., MBA, director of medical computing at Kitware, Inc. (http://www.kitware.com)
13. **How the FDA Reviews Software and Related Issues**: Dr. Nicholas Petrick, deputy director Division of Imaging, Diagnostics and Software Reliability, Office of Science and Engineering Laboratories (OSEL), Center for Devices and Radiological Health (CDRH), U.S Food and Drug Administration (FDA) -- http://www.fda.gov
14. **Machine Learning and Software Engineering**: Dr Christian Kastner, associate professor at the Institute for Software Research, Carnegie Mellon University, Pittsburgh, PA. See this page for more information.

Recommended Books

- **Medical Device Software: Verification, Validation, and Compliance**, David A. Vogel
- **Safeware: System Safety and Computers. Addison-Wesley**, Nancy G. Leveson
- **Medical Device Cybersecurity for Engineers and Manufacturers**, Alex Wirth, Christopher Gates, and Jason Smith
- **Development of FDA-Regulated Medical Products: Prescription Drugs, Biologics, and Medical Devices**, Elaine Whitmore
- **Hacking Healthcare: A Guide to Standards, Workflows, and Meaningful Use**, Fred Trotter and David Uhlman
- **Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide**, Oleg S. Pianykh
- **Open Source Licensing: Software Freedom and Intellectual Property Law**, Lawrence Rosen
- **fMRI Neurofeedback**, Michelle Hampson
- **Probability, Random Variables and Stochastic Processes**, Athanasios Papoulis and S. Unnikrishna Pillai
- **The Book of Why: The New Science of Cause and Effect**, Judea Pearl and Dana Mackenzie
- **Science Fictions: How Fraud, Bias, Negligence, and Hype Undermine the Search for Truth**, Stuart Richie
- **Deep Learning**, Ian Goodfellow, Yoshua Bengio and Aaron Courville
- **Disciplined Entrepreneurship**, Bill Aulet
- **Talking to Humans: Success Starts with Understanding your Customers**, Giff Constable
- **The American Health Care Paradox: Why Spending More is Getting us Less**, Elizabeth Bradley and Lauren A. Taylor