Measuring Happiness: http://www.hitachi.com/rev/pdf/2015/r2015_08_116.pdf

- Happy People 37% more productive than unhappy people at work (97)
- 300% increase in creativity (97)
- Large # of happy people leads to higher earnings per share
- Card measures activity ---
- Happy points based off of concentrations, enjoyment, good sleep, desires, conversation, appetite, depression, anxiety, loneliness and sadness

Social Tech in Business:
http://www.nextlearning.nl/wp-content/uploads/sites/11/2015/02/McKinsey-on-Impact-social-technologies.pdf

- Social technologies used to communicate with 38% of customers
- Mobile enterprise-- 48% of company have access
- Cloud as delivery platform-- cost effective, confidentiality?
- 20% increase in revenue because of social tools
- 18% increase in cost improvements
- Employees could save 30% of time if using searchable, social tech instead of stuff like email

How have Software Engineering been measured?
http://www.scitepress.org/Papers/2017/63144/63144.pdf

- Software engineering provides info on selected objs and events-- make understandable and controllable
- Necessary for any kind of improvement
- Productivity = effectiveness of productive effort measure in terms of rate of output/unit of input

Development Matrices to look at:
https://blog.gitprime.com/5-developer-metrics-every-software-manager-should-care-about/

- Lead Time: time between beginning of a projects development and delivery to customer; help predict when product is ready with accuracy
- Churn: percent of developers own code representing and edit to their recent work;
    - measured in lines of code that have been modified/added/deleted;
    - used to control quality of software engineering process
    - High churn = something is wrong
- Impact: measure of effect that code changes have on project
    - Difference between adding more lines of code to adding files
- Active Days: day when engineer contributed code to project, including specific tasks like writing and reviewing code
    - Non engineering parts- planning, meetings, chasing down specs (non-active day)
- Efficiency: percentage of an engineer's contributed code that's productive
    - Involves balancing coding output against longevity
    - High churn reduces efficiency and business value
    - Independent of amount of code

https://devops.com/software-development-better-way-measure-success/

- Controls can result in unintended consequences
- Drop regressions with incremental features→ lower quality
- Choose goals and select control metrics
- Levels of monitoring situations to schedule time for checking in
- Find goals that work for your need then build metrics to keep everyone accountable

9 Metrics that make a difference in Software Engineering:
https://techbeacon.com/9-metrics-can-make-difference-todays-software-development-teams

- Agile Process-
    - aid in planning and inform decisions about process improvement
    - Lead time, cycle time, team velocity, open/close rates
    - Help find which processes need attention
    -
- Production Analytics
    - Application crash rate-- # of times application fails/# times it was used
- Security metrics
    - Used in build process
    - Be mindful
    - Ex:
        - Endpoint incidents- how many endpoints have experience virus over period of time
        - Mean Time to Repair- time between discovery of breach and breach resolution
- Success = ultimate metric
    - Properly quantified questions imply metrics
    - Metrics should only be used to answer questions
    - Metrics used to learn only as long as they prove useful for driving improvements

What do they measure and how do we know?
http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=96403470F6E0EF15C14F48EE2566B3A5?doi=10.1.1.1.2542&rep=rep1&type=pdf

- Measurement is the assignment of numbers to objects or events according to rule
- Measurement is the process of empirical, objective, assignment of numbers to properties of objects or events of the real world in such a way as to describe them
- Set of Metrics:
    - Correlation- metric should be linearly related to quality factor as measured by stats between metric and corresponding quality factor
    - Consistency
    - Tracking
    - Predictability
    - Discriminative Power

    - Reliability
- Direct Metric = a metric that does not depend upon a measure of any other attribute; presumed valid and other metrics are validated in terms of it; one variable
- Derived Functions- Programmer productivity, module defect density, requirements stability, system spoilage
- Direct measurements (length of source code, duration of testing process, number of defects discovered, time programmer spends on project,) all intrinsically complex
- Construct validity- How do you know that you are measuring what you think you are measuring?
- Measurement is the empirical, objective assignment of numbers, according to a rule derived from a model or theory, to attributes of objects or events with the intent of describing them
- Key release criteria- acceptably low count of significant, unfixed bugs
- Bug curves- graphs of how many new bugs are discovered week by week, or how many bugs are unresolved week to week

For Engineering Today, Stop Measuring Productivity:
http://theworkspacetoday.com/2017/01/13/for-engineering-performance-stop-measuring-productivity/

- Bigger isn't always better
- Value of work > cost of work
- Incremental improvements can deliver more value faster
- Customers view product as continuum not separate pieces
- Build things that make your job easier
- See why customer thinks it's a requirement

You can and Should Measure Software Engineering Performance
http://engineering.kapost.com/2015/08/you-can-and-should-measure-software-engineering-performance/

- Productivity-
    - Lines of code-- not good measurement cause it's better to reduce than increase lines of code
    - Dates-- hitting deadlines is fine but can sacrifice quality of code
    - High level iteration goals based on velocity
- Business Impact:
    - Inc # of users
    - Driving better adoption of system
    - Engineering team works with product team
- Bugs-- measure volume and severity
- Code Quality
    - Test Coverage
    - Code Quality-- ie codeClimate to give grade on repositories; what changes can be made before merging code
- App Performance
    - Use Apdex which measures user satisfaction

- Use all metrics and blend scores to see what you want to focus on in each time period

IEEE Computer Society-- Code of Ethics: https://www.computer.org/education/code-of-ethics

- Software Engineering Code of Ethics and Professional Practice (SECEPP)
- Software Engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession
- Follow 8 Principles:
    - PUBLIC - SE shall act consistently with public interest
    - CLIENT AND EMPLOYER- act in manner that is in best interests of their client and employer consistent with public interest
    - PRODUCT- ensure their products and modifications meet highest professional standards possible
        - ******LOOK AT APPLE
    - JUDGMENT- maintain integrity and independence in their professional judgment
    - MANAGEMENT- subscribe to and promote an ethical approach to the management of software development and maintenance
    - PROFESSION- Advance integrity and reputation of profession consistent with public interest
    - COLLEAGUES-- Be fair to and supportive of their colleagues
    - SELF- SE shall participate in lifelong learning regarding their practice of their profession and shall promote an ethical approach to the practice of the profession
- In situations where standards and the code conflict, SE should use ethical judgment to act in a manner which is most consistent given circumstances

Ethical Framework: file:///Users/kwalsh/Downloads/10.1007_s11948-015-9665-x.pdf

- Software Crisis: coined to express the gap between ability to systematically develop ‘quality’ software product (correct, understandable, reliable, stable, verifiable) and the rapid expansion of computing power
- Over time, only ⅓ of software development projects end successfully and on time
- 15% projects fail almost immediately with the stat of developments
- ½ projects run into problems and deviate significantly from estimated budgets and schedules
- Lots of post delivery changes for successful projects
- No silver bullet solution (BROOKS)
- SOLUTION-- ethical framework
    - Engineers interconnectedness with clients and other stakeholders to deal with challenges and failures in software development process
    - Ethical Driven Software Development
- Engineering Definitions:
    - Creative application of scientific principles… all respects an intended function, economics of operation and safety to life and property
    - Branch of science and technology concerned with the design, building and use of engines, machines and structures

    - Field of study or activity concerned with modification or development in a particular area
- Software Engineering- application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; application to engineering to software
- Many users don't understand application/software package but have dependence on computers and tech → end-user/consumer dependence on software professional
- Must be able to fully control software user is dependent on
- Tech that controls car warnings→ if something bad happens, engineer responsible
- Ethical framework connected to the day to day professional activity of software engineers and development process
- Ethics = inspirational normative ideal without practical bearing ATM
- Declaration of Helsinki- ethical guideline that requires the physician to follow certain protocol in research design,
- Phases of Software Development
    - Initiation Phase
    - Requirement Phase
    - Design Phase
    - Development Phase -- not informed consent; software engineer needs to confirm client understands what's at stake
    - Testing and maintenance Phase
- ESDS Index/Questionnaire for determining ethical decisions
- Approach is a process

Apple and FBI:
https://www.nytimes.com/2016/03/30/technology/apples-new-challenge-learning-how-the-us-cracked-its-iphone.html

- Apple's ethical responsibility to fix hacker situation and make sure it can't happen again
- FBI's ethical responsibility to bring justice-- hence hacking
- Where's the line between keep security and ensuring security for users?
- Ethics aren't black and white
- FBI figured out how to hack the iPhone without any help from Apple
- FBI won't tell Apple how they did it, therefore Apple can't ensure that it won't happen to other, non-criminal users

Ethics in Software Engineering: Kamthan
https://pdfs.semanticscholar.org/caa6/07b7a74db0a71fd66b21fe0c8643fad7e3f7.pdf

- Ethics = code of professional standards, containing aspects of fairness and duty to the profession and the general public
- Some tech failures lead to death (ie in cars) others just inconvenience
- Lack of specific guidance for improvement of software quality within the domain of software ethics
- SECEPP has issues:
    - Lack of separation of concerns
    - Recency

    - Precision
    - Completeness
    - Reachability to certain audiences
    - Specificity→ makes realization difficult
-