“Evidence-based practices in software development”

weblog.raganwald.com

"Evidence-based medicine de-emphasizes intuition, unsystematic clinical experience, and pathophysiologic rationale as sufficient grounds for clinical decision-making"

Evidence-Based Medicine: A New Approach to Teaching the Practice of Medicine

Manifesto for Evidence-based practices

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Unbiased observations over anecdotes and authority
Original results over interpretations and summaries
Diagnosis and Inspection over instinct and experience
Managing uncertainty over singular prognosis

That is, while there is value in the items on the right, we value the items on the left more.

What's wrong with the stuff on the right?

- Authority and anecdotes are notoriously bad.
- Summaries and interpretations are made in the context of someone else, somewhere else, working on something else. YMMV!
- Instinct and experience are based on something else, somewhere else. Times change!
- Nothing in software is certain, software is more like quanta than celestial bodies.

How do evidence-based practices help?

- Lose weight
- Grow your hair back
- Lose the hair on your back
- Look better
- Make more money
- Power and influence over your colleagues

"Systematic attempts to record observations in a reproducible and unbiased fashion markedly increase the confidence one can have in knowledge about patient prognosis, the value of diagnostic tests, and the efficacy of treatment."

Evidence-Based Medicine: A New Approach to Teaching the Practice of Medicine

How do evidence-based practices help?

Confidence.

Evidence-based practices increase our confidence in our own predicted outcomes, they increase our confidence in our decisions, and they increase our ability to communicate that confidence within and outside the team.

The scope of this discussion:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Unbiased observations over anecdotes and authority
Original results over interpretations and summaries
Diagnosis and Inspection over instinct and experience
Managing uncertainty over singular prognosis

That is, while there is value in the items on the right, we value the items on the left more.

What is evidence?

"An objective input to a function giving the likelihood of achieving a precisely defined outcome."

“Evidence: an objective input to a function giving the likelihood of achieving a precisely defined outcome.”

“Evidence: an objective input to a function giving the likelihood of achieving a precisely defined outcome.”

One of these things is not like the others…

- Database of defects managed by QA
- Memo analyzing source code after a review
- JProbe Threadalyzer results
- Unit test results
- Compliance with (a) J2EE standards, (b) GoF patterns, or (c) goto-less code

Some tests of objectivity

- Is the evidence reproducible in time?
  - Perform the same ‘experiment’ again, do you get the same results?
- Is the evidence reproducible in ‘place’?
  - If someone else performs the same ‘experiment’, do you get the same results?

One of these things is not like the others*...

1. Unit test results
2. JProbe Threadalyzer results
3. Compliance with (a) J2EE standards, (b) GoF patterns, or (c) goto-less code
4. Database of bugs managed by QA
5. Memo analyzing source code after a review

“

Evidence: an objective input to a function giving the likelihood of achieving a precisely defined outcome.”

"There are three kinds of lies: lies, damn lies, and statistics."

Benjamin Disraeli

What is the difference between statistics and evidence?

Statistics:

The practice, study or result of the application of mathematical functions to collections of data in order to summarize or extrapolate that data.

Evidence:

A thing or things helpful in forming a conclusion or judgment.

The functional perspective

- A function takes inputs and gives an output that varies only with the inputs.*
- The inputs are your evidence.
- The output is your conclusion.

If your conclusion varies given the same inputs, you do not have a function, or you are hiding one of the inputs.

“Evidence: an objective input to a function giving the likelihood of achieving a precisely defined outcome.”

"The results of the relevant study show that the patients risk of recurrence at one year is between 30% and 43%, and at three years is between 51% and 60%. After a seizure-free period of 18 months his risk of recurrence would likely be under 20%. She conveys this information to the patient."

Evidence-Based Medicine: A New Approach to Teaching the Practice of Medicine

“Evidence: an objective input to a function giving the likelihood of achieving a precisely defined outcome.”

One of these things just doesn't belong…

- The design shall be object-oriented.
- The server shall not leak memory.
- Reg will be done coding the workflow manager on or before July 31, 2003 .
- The customer will accept the software on or before January 16, 2004.
- The software will be defect-free.

One of these things just doesn't belong*...

1. The design shall be object-oriented.
2. Reg will be done coding the workflow manager on or before July 31, 2003 .
3. The customer will formally accept the software on or before January 16, 2004.
4. The software will be defect-free.
5. The server shall not leak memory.

“Evidence: an objective input to a function giving the likelihood of achieving a precisely defined outcome.”

“Hard data about what will be ready by when, and why.”

What is an evidence-based practice?

1. Generates/gathers evidence
2. Forms a conclusion or judgment about the potential outcomes
3. Takes action based on the evidence to maximize the likelihood of a desirable outcome

What is an evidence-based practice?

- Generates/gathers evidence, e.g.
  - How many stories overrun per iteration?
  - How many regressions per iteration?
  - What is the complexity of each source file?

What is an evidence-based practice?

- Forms a conclusion or judgment about the potential outcomes, e.g.
  - Relative estimates (jellybeans) are sound, but absolute estimates (days/jellybean) are low, therefore project is losing velocity.
  - Some code has become highly coupled, therefore velocity will continue to drop.
  - Some code has become too fragile, therefore converging defects will be high uncertainty.

What is an evidence-based practice?

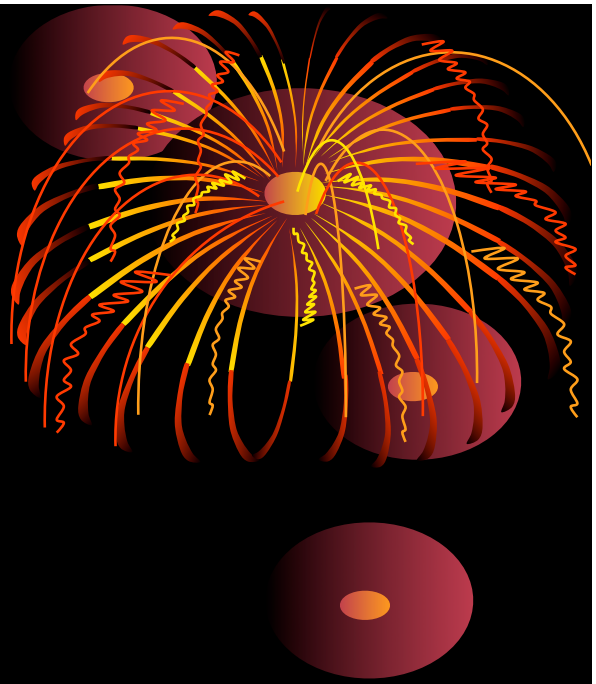
- Takes action based on the evidence to maximize the likelihood of a desirable outcome, e.g.
  - Trim number of stories per iteration? Adjust scope of stories?
  - Refactor ‘hotspot’ components to lower coupling?
  - Increase test coverage of complex components?

The evidence feedback loop

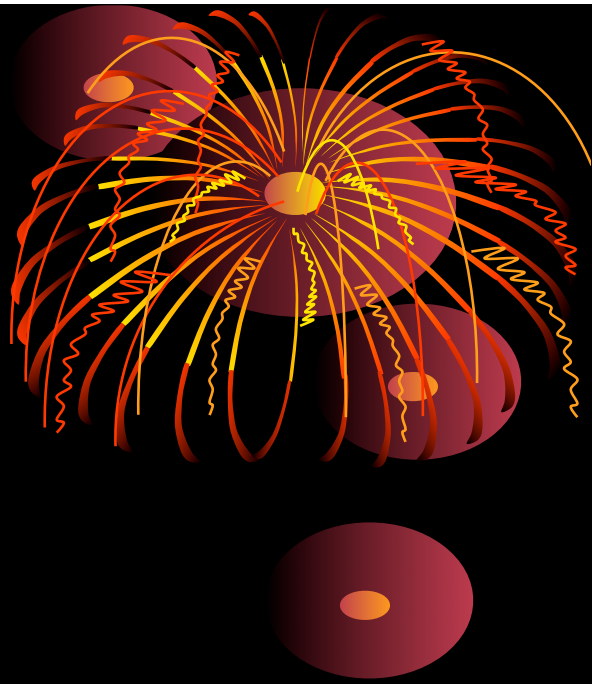
Evidence-based software development practices

- It’s all about confidence and managing uncertainty.
- Objective input + function = probability of well defined outcome.
- Then take action to turn evidence into a practice
- Repeat, creating a feedback loop

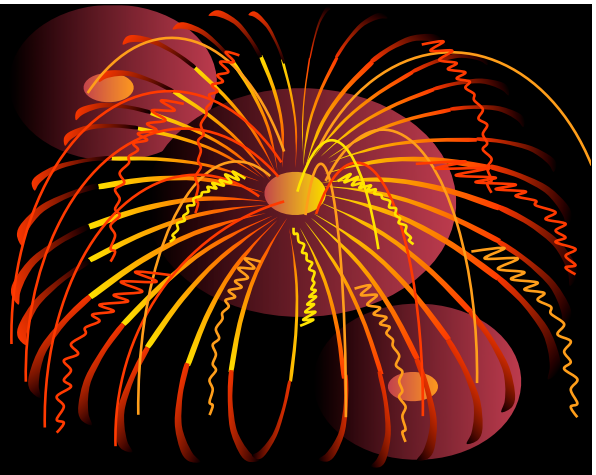
Pop Quiz!



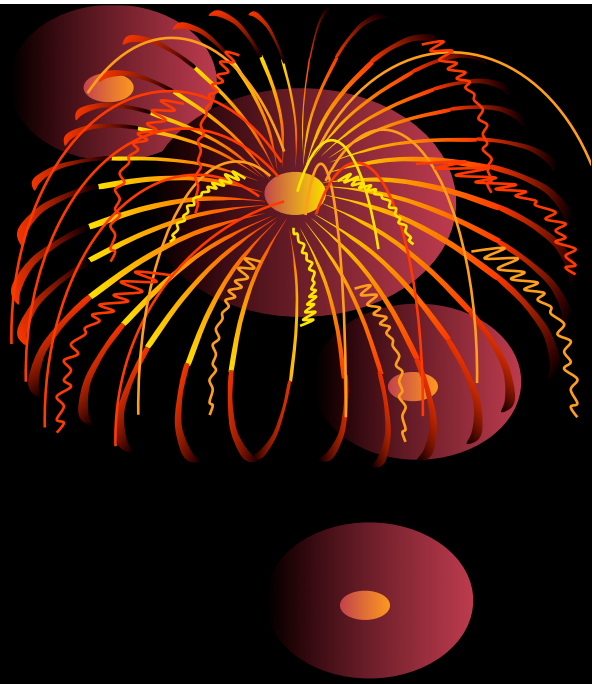
Which of the following statements are reflective of evidence-based practices for inspection, diagnosis, and managing uncertainty?



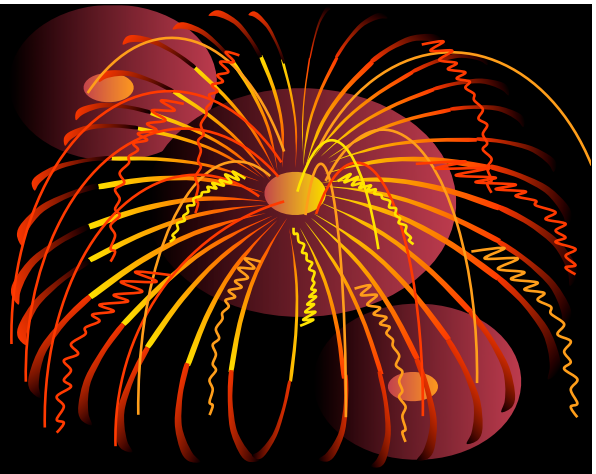
“Unit tests are released into the code repository along with the code they test. Code without tests may not be released.”



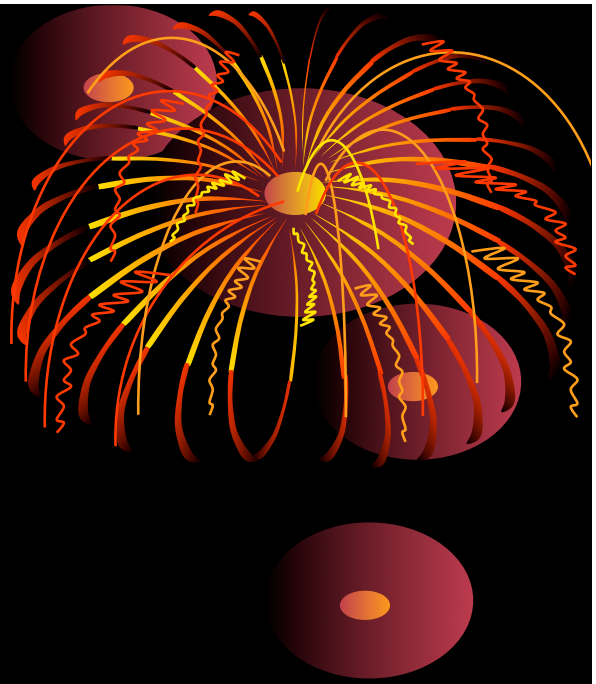
“Pair programming increases software quality without impacting time to deliver. It is counter intuitive, but 2 people working at a single computer will add as much functionality as two working separately except that it will be much higher in quality.”



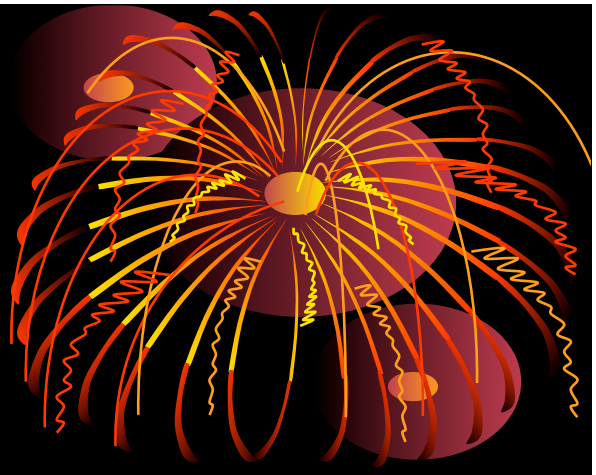
“Evidence shows that programmers are more productive creating business applications with Smalltalk than they are with Cobol.”

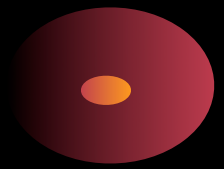


“When you create your tests first, before the code, you will find it much easier and faster to create your code… You also have immediate feedback while you work. It is often not clear when a developer has finished all the necessary functionality. Scope creep can occur as extensions and error conditions are considered. If we create our unit tests first then we know when we are done; the unit tests all run.”



“Developers should be integrating and releasing code into the code repository every few hours, when ever possible.”





“JProbe Memory Debugger revealed that our Swing application creates many lapsed listeners, leaking memory. We fixed the problems and run the debugger every few iterations just in case.”

"So little done, so much to do."

Cecil John Rhodes (1853-1902)

Connecting Outcomes to Project Success

Nobody wants to report that the operation was a success but nevertheless the patient died.

Partial Outcomes

- Outcomes like “requirements gathered, “design complete,” and “code complete,” are partial indicators of project success
- So are outcomes like “June sprint successful,” “unit tests passing,” and “velocity restored to previous average”

It doesn’t matter what methodology you use: you construct a successful project out of successful partial outcomes

What good are evidence-based practices?

- Provide confidence in our ability to achieve successful partial outcomes
- Provide an objective basis for communicating status and making decisions

But… evidence-based practices cannot select the best partial outcomes by themselves. They are ‘necessary but not sufficient’.

“The world is divided into two kinds of people: those who bifurcate and those who don’t.”

How to select partial outcomes

Divide all possible partial outcomes into two piles: those that reflect the necessary ends from those that reflect your preferred means.

Ends are better than means (although means are better than nothing at all).

Ends and Means

- Customer value
- Correctness
- Done-ness of final software*

- Velocity
- Standardization
- Done-ness of infrastructure*

While there is value in the items on the right, we value the items on the left more.

Done-ness vs. Done-ness

- Done-ness of the final software is some property of the finished software you can inspect.
- Done-ness of infrastructure is completion of some intermediate product that does not evidence any completion of the final software.

Back to Ends and Means

- Means seem necessary but never sufficient for project success.
- Ends are sufficient in and of themselves.

Delivery of value is the ultimate end. Isn't that how you measure project success?

If you can't deliver value incrementally, done-ness of the final software is a decent proxy.

Evidence-based software development practices

- It's all about confidence and managing uncertainty.
- Objective input + function = probability of well defined outcome.
- Need to then take action to turn evidence into a practice
- Prefer outcomes that reflect 'ends' over outcomes that reflect 'means'

"Friends applaud, the comedy is over."

Ludwig van Beethoven (1770-1827)

Supercharging Evidence-Based Software Development

Turning a tactic into a strategy,
or,
How I Learned to
Stop Worrying and Love XP

"The best strategic plan is useless if it cannot be executed tactically. "

Field-Marshall Erwin Rommel

quoted in "Bottom-Up Marketing"

Ends, Means, and Outcomes (The Strategic Approach)

1. Define the desired end result. Often this is one monolithic outcome.
2. Select means for achieving the outcome based on minimizing time and cost.
3. Organize the team's practices around outcomes that reflect the means.

Bottom-Up Thinking

- The tactic is the practice that produces results
- The strategy is the organization of the development team's activities to produce the maximum tactical pressure

Our tactic: Evidence-based practices

1. Generate/gather evidence
2. Form conclusions or judgments about the potential outcomes
3. Take actions based on evidence to maximize the likelihood of desirable outcomes
4. Create a feedback loop

Result: confidence and the ability to manage uncertainty.

The evidence feedback loop

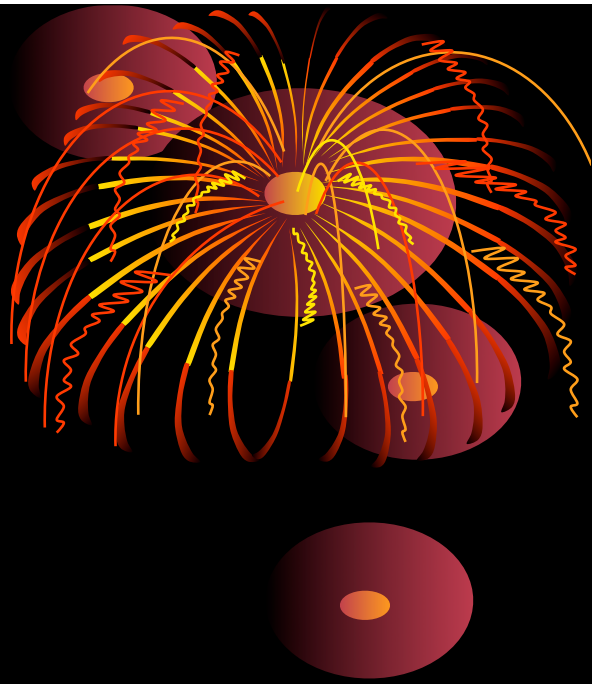
Bottom-Up Evidence-Based Software Development

- Our tactic is the evidence-based feedback loop.
- Our strategy is organizing the team's activities around maximizing the quantity and quality of evidence-based feedback loops.

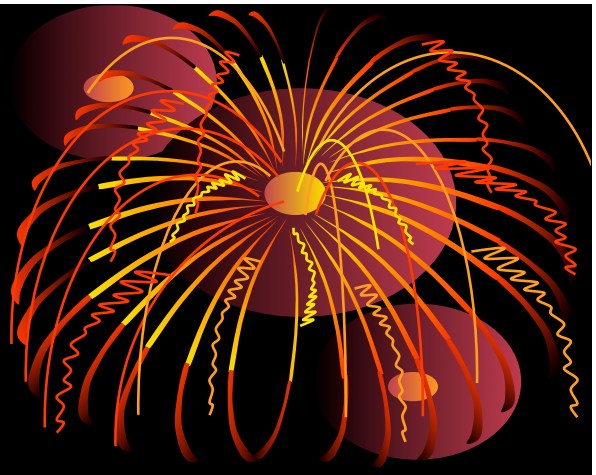
Ends, Means, and Outcomes (The Bottom-Up Approach)

1. Define the desired end result in such a way that you can divide it into many partial ‘end’ outcomes.
2. Select means for achieving the outcome based on maximizing the quantity and quality of feedback loops.
3. Organize the team’s practices around the feedback loop.

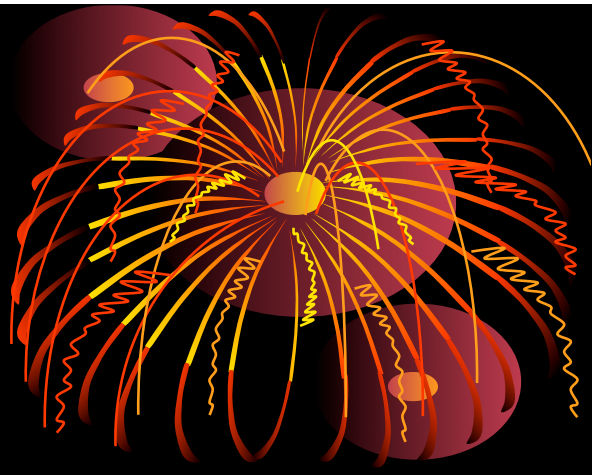
Final Exam



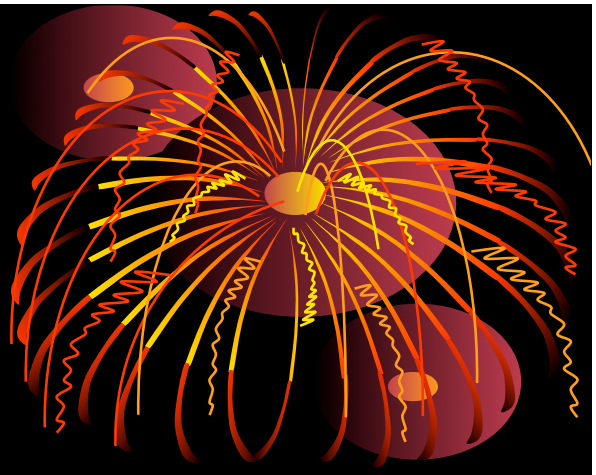
Which of the following practices organize the team's activities around maximizing the evidence feedback loop?



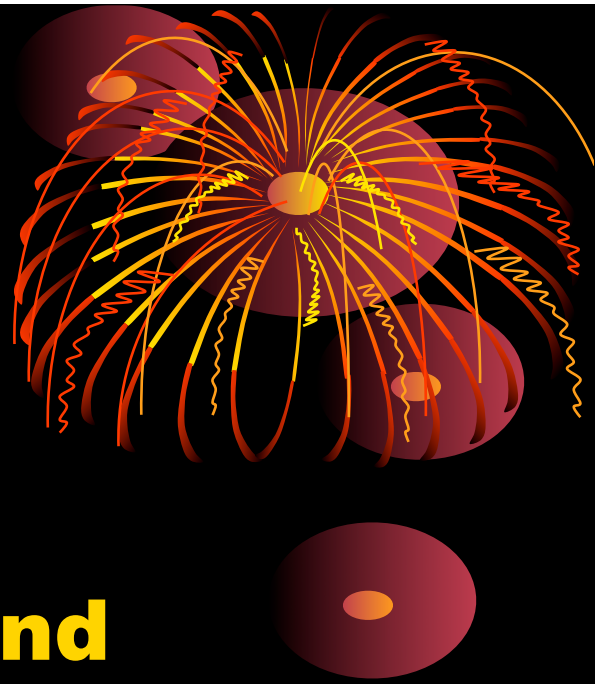
“The development team needs to release iterative versions of the system to the customers often.”



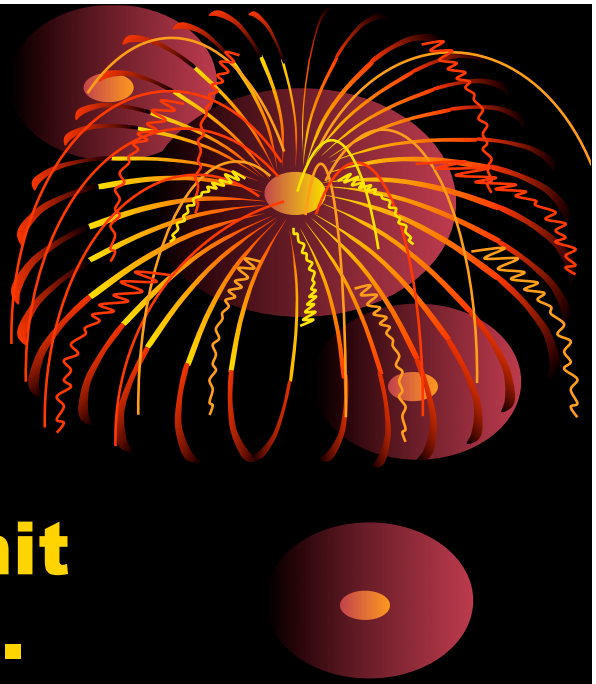
“Move people around to avoid serious knowledge loss and coding bottle necks.”



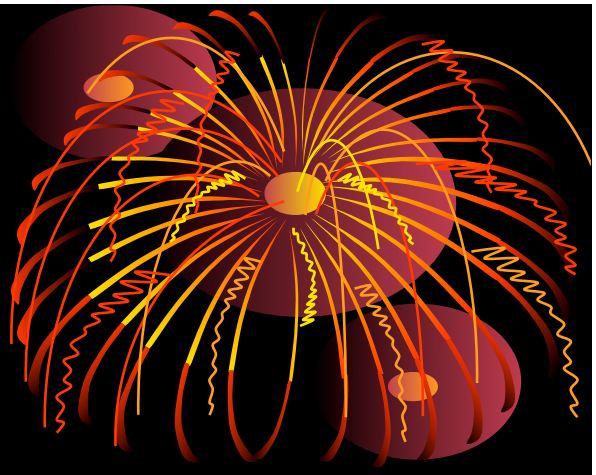
“The release planning meeting is used to discover small units of functionality that make good business sense and can be released into the customer's environment early in the project.”

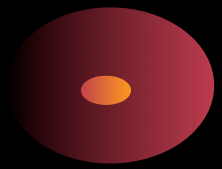


“Use a release planning meeting to re-estimate and re-negotiate the release plan if your project velocity changes dramatically for more than one iteration.”



“It is often very difficult to unit test some software systems. These systems are typically built code first and testing second, often by a different team entirely. By creating tests first your design will be influenced by a desire to test everything of value to your customer. Your design will reflect this by being easier to test.”





“Refactor mercilessly to keep the design simple as you go and to avoid needless clutter and complexity. Keep your code clean and concise so it is easier to understand, modify, and extend.”

"Bring down the curtain, the farce is played out. "

Francois Rabelais (1494?-1553)

uni
Merci n
va ko; Wabeeja
anta; Asante; Aksante
la mó; Shukria; Maigo; Ma
amnida; Komawoyo; Kulo; Kulo m
Dot nuet; Keyi tapon; Shukur; Sipas; Si
ula; Mèsi; Mèsi plen; Mèsi anpil; Granmèsi; Od
gracias; Ah bo; Ah bo u ja; Aw bon uija; Da ja; Òboi jâ,
wp jai; Khawp jai lai lai; Gratia; Gratias; Gratias tibi ago; Gra
ok chi; Melesí; Matóndo; Natondi yo; Gratzia; Mouchou gratzia; Xu
umu; Achiu; Dekoju; Labai achiu; Dekui; Labai dekoju; Nuoširdziai deko
ke; Ek dank auk schoin; Sind auk viellmaols bedankt; Moducué; Webale; Web
qe siam; Kusakililaku; Webale; Webale; Erokamano; Ck'wálidxw; Helí'dubshewx; Gu
Aske; Ashi; Ashi oleng; Ashi naling; Eso; Blagodaram; Asantte; Kooshukhuru; Marahaba; M
tompoko; Misaotra indrindra; Terima kasih; Terima kasih banyak-banyak; Nandi; Valarey nan
i ke; Grazzi; Grazzi hafna; Chjóonte; Chjoonta; Chjónta tey; Chjónta che; Sanco; Mossi; Ban
e; Abaraka; I ning bara; Gura mie ayd; Gura mie eu; Gura mie mooar ayd; Mauruuru; Tika ho
aki; Meitaki ma'ata; Krasia may; Chaltu; Chaltu may; Traeltu; Manumeimi; Abhari ahi; Dhanya
vawaatha; Tau; Koutai; Kommol; Koko; Pöjö; Natejchiri; Nkhi k'a ninashitechino; Baiika; Bisse
Laengz zingh; Laengz zingh meih; Laengz zingh camv; Tö' dun; Welálin; Weláliek; Akpe; Ak
asih; Tingki; Tengkiju; Kúta'ùná; Kúta'ùrí; Kúta'ù shããrí nuùro; Niku tab'i; Ta xa'u zin; Cuta
ixensa; Tyáhvi nyóò; Nihedebil; Wneeweh; Oneowe; Multumesc; Tang kun; Merçì; D
Tand ikh bayarlalaa; Saikhan zochluullaa; Ta ikh tus bolloo; Wenatase; Barka
o; B'o'tic; Syukprya; Mersi; Tanikiu; Mèu mèu; Kewa; Mvto; Henka; Ka
ncamati huel miac; Tlazohcamatzin; Tlazocama; Tlazocamatl; Tl
Kongoi; Kaigai; Asai; Gràzzie; Tobotonoque; Ttaubotnean
seiq; Ngeyabonga; Ngiyabonga; Ngiyathokaza; Ng
onga kakulu; Gaantangi fi ye; Gaantangi; A bi
si; Nägê; Dé kãã; Ná goodoota-ngaa n
Ha ia; Fakaaue lahi mahaki; Tak
haa kennu; Maharaba; Ga
Losaka; Hihuri; Nam
ng; Ke kmal