Software Estimation

also known as

Black Magic

Estimate Sanity Check

1. Was a standard procedure used to create the estimate?
2. Was the process free from pressure that could bias the result?
3. If the estimate was negotiated, were the inputs negotiated and not outputs or process?
4. Is the productivity assumption underlying the estimate comparable to productivity actually experienced on past projects of similar size?

Estimate Sanity Check

5. Is the estimated schedule at least
   `2 * ( StaffMonths ^ 0.34 )`

6. Did the people who are going to do the work, do the estimation?

7. Does the estimate include a non-zero allowance for impact of project risks?

8. Is the estimate a part of a series of estimates that will be done across the life of the project?

9. Are all elements included, for eg: creating infra, setup git, migration to new system, upgrading tooling etc.?

Estimate Sanity Check

- What the scores probably mean…
  - *7-9 — highly accurate*
  - *4-6 — probably optimistic*
  - *0-3 — nearly useless*

Estimates vs Targets

- "How much time (kinda) will it take to finish all these things?
- Estimates always have a probability

Estimates vs Targets

- Specific timelines that must be met
- The timeline is usually immovable
  - *Exhibition, investor meeting etc.*

Artificial Pressure

- Sometimes the pressure is internal
- Do NOT underestimate!
  - *Underestimation is **worse** than Overestimation!*

Parkinson's Law

- The idea that work will expand to fill the available time
- This is why some managers squeeze estimates

Goldratt's Student Syndrome

- Given too much time, developers will procrastinate until late in the project
- Then they will rush towards the end, and not complete things on time

No Underestimation!

- Statistically reduced chance of on-time completion
- Engineers are too optimistic to begin with

No Underestimation!

- Increased chance of poor technical foundation, worse in the long run
- Destructive late-project dynamics and behaviour

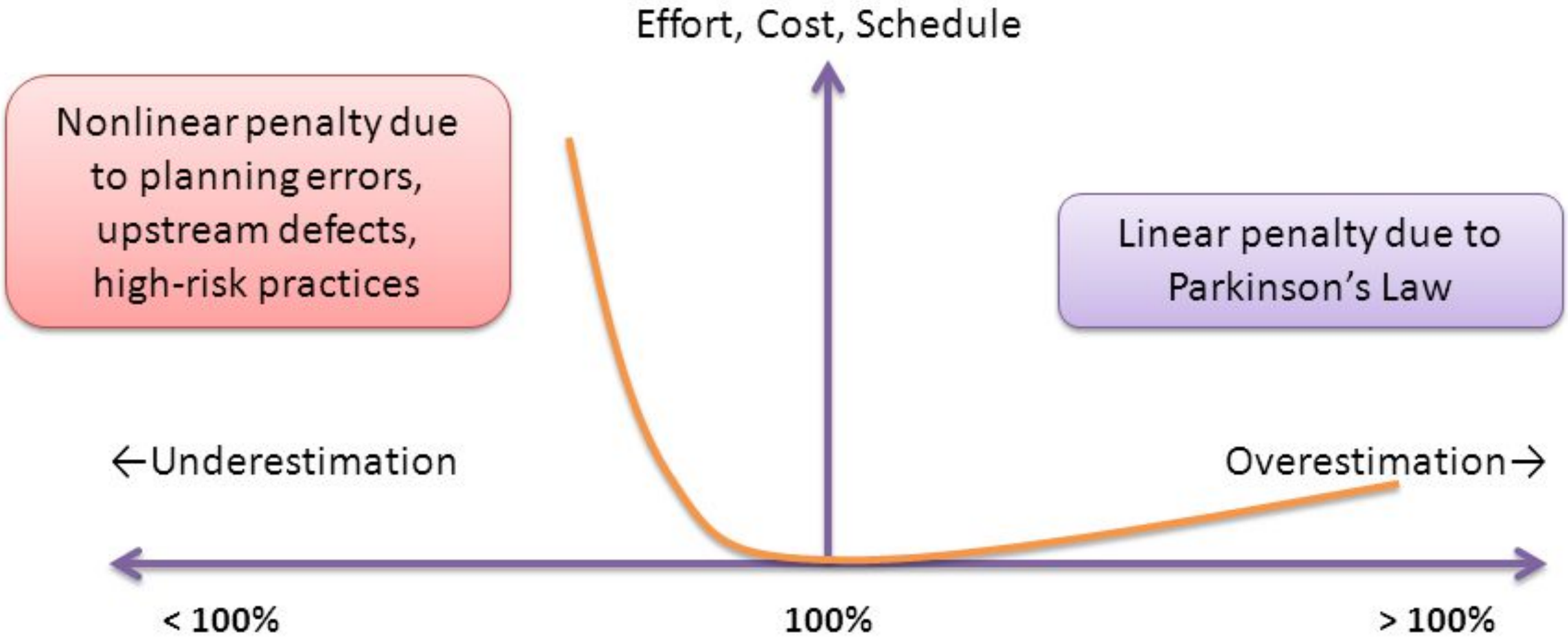
No Underestimation!

- *... more status meetings with management*
- *... frequent re-estimation*
- *... preparing interim releases to increase customer confidence*

No Underestimation!

- *... more meetings to cut scope*
- *... fixing problems arising from quick and dirty hacks done earlier*

Accuracy of Estimates

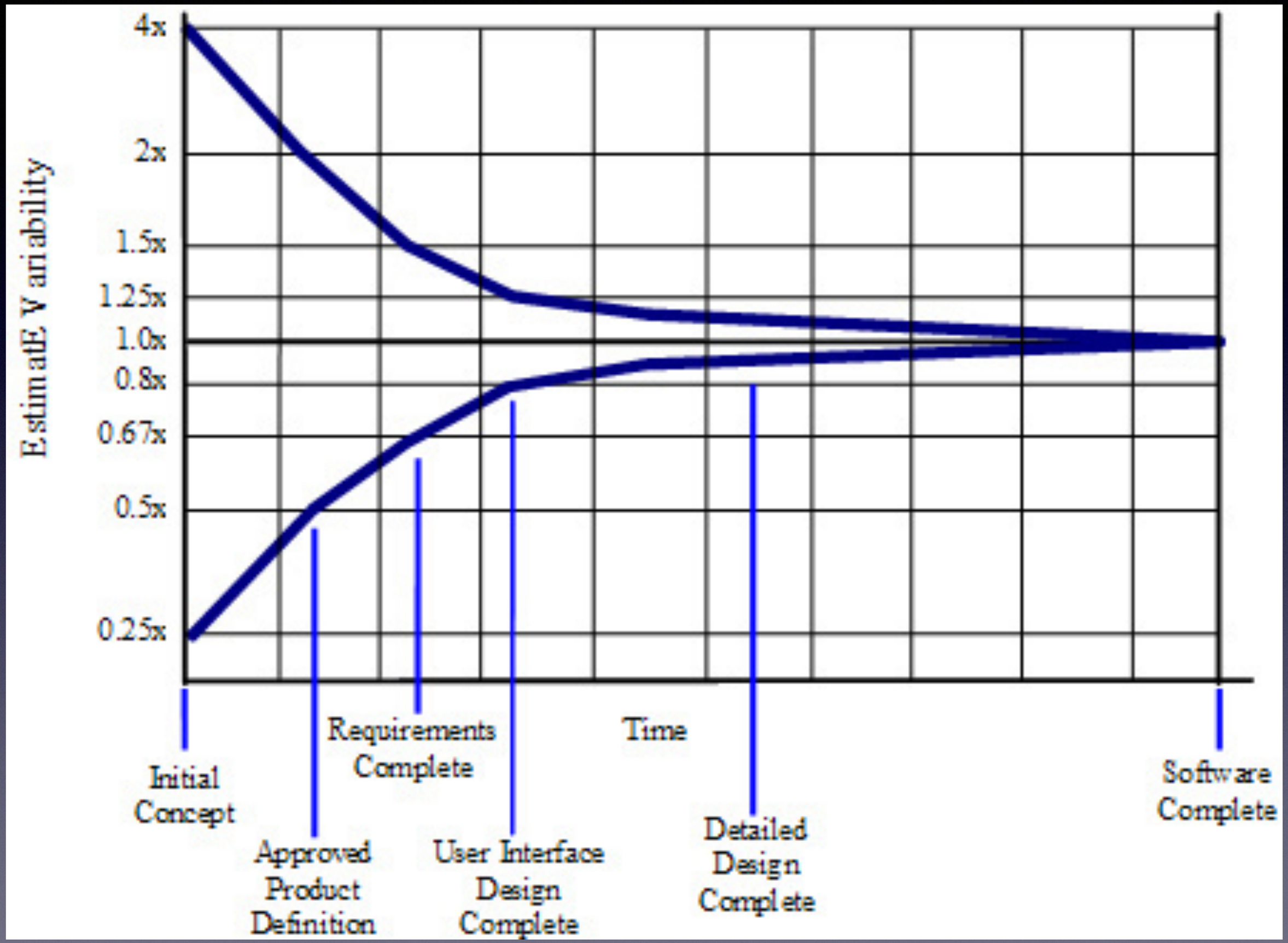


Penalties of underestimation more severe than those for overestimation. If you can't estimate with complete accuracy, it's better to err on the side of overestimation

– Steve McConnell

Cone of Uncertainty

- Estimates become more accurate as the project progresses
- *Yes, that is quite obvious.*

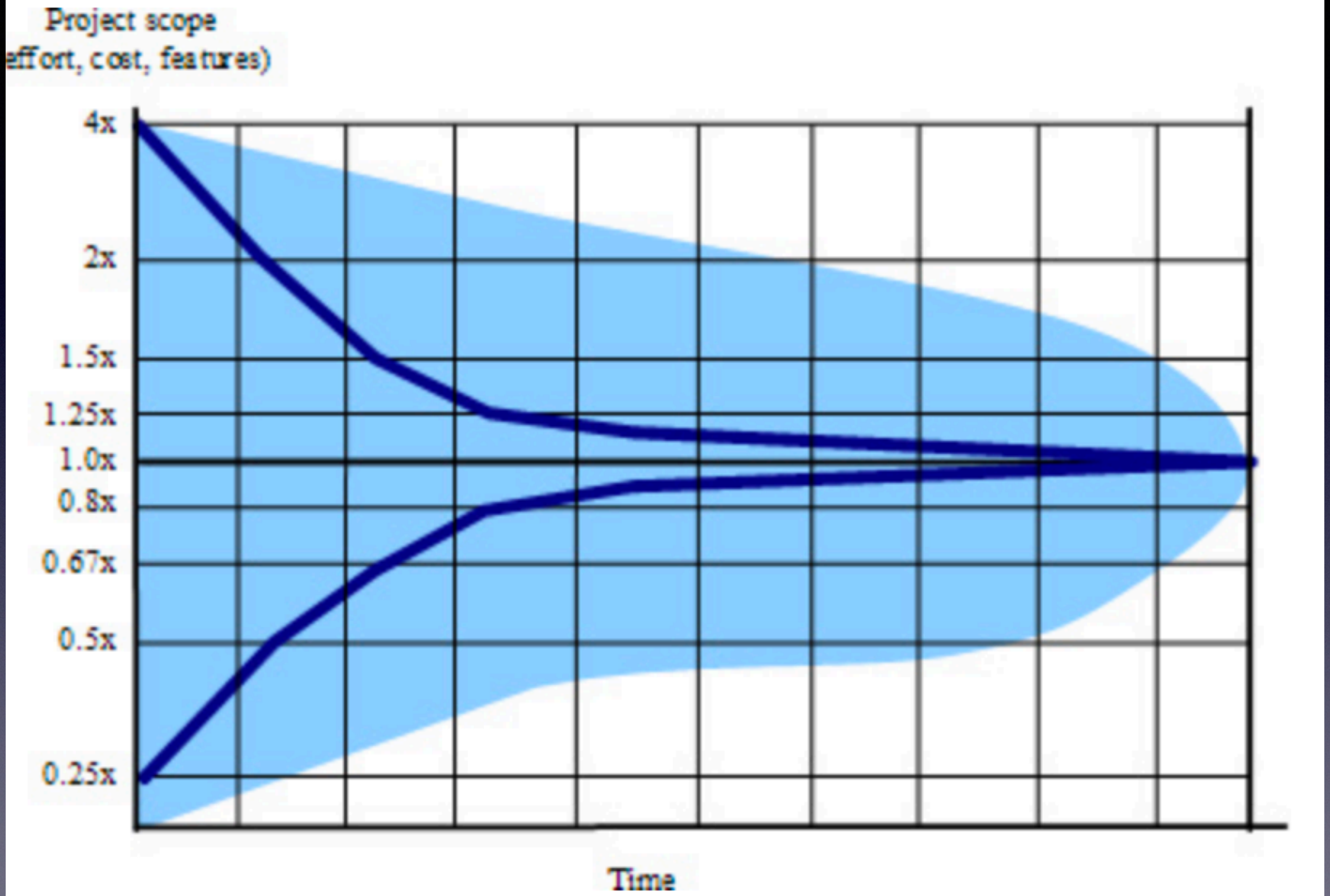

4x
2x
1.5x
1.25x
1.0x
0.8x
0.67x
0.5x
0.25x
Estimate Variability
Time
Initial Concept
Approved Product Definition
Requirements Complete
User Interface Design Complete
Detailed Design Complete
Software Complete

Cone of Uncertainty

- The Cone represents the best-case accuracy that is possible
  - *… as if created by a Skilled Estimator*
- It is **easily** possible to do worse, but not better

Cone of Uncertainty

- The Cone does not narrow itself
- Narrow the Cone by removing sources of variability in the project


Project scope
effort, cost, features)
4x
2x
1.5x
1.25x
1.0x
0.8x
0.67x
0.5x
0.25x
Time

Cone of Uncertainty

- Include all the associated engineering activities in your estimates
  - *… stated requirements*
  - *… implied requirements*
  - *… non-functional requirements*

So many requirements!

- Deployment
- Maintaining build scripts
- Technical reviews
- Creating test data
- Writing documentation
- Ramping up new team members
- Tuning performance
- Upgrading tooling and systems

No off-the-cuff!

- Off-the-cuff estimates are the very worst, and set the worst expectation
  - *It's worse for you, the developer!*
- Even a 15 minute estimate is more accurate than one pulled out of THIN AIR

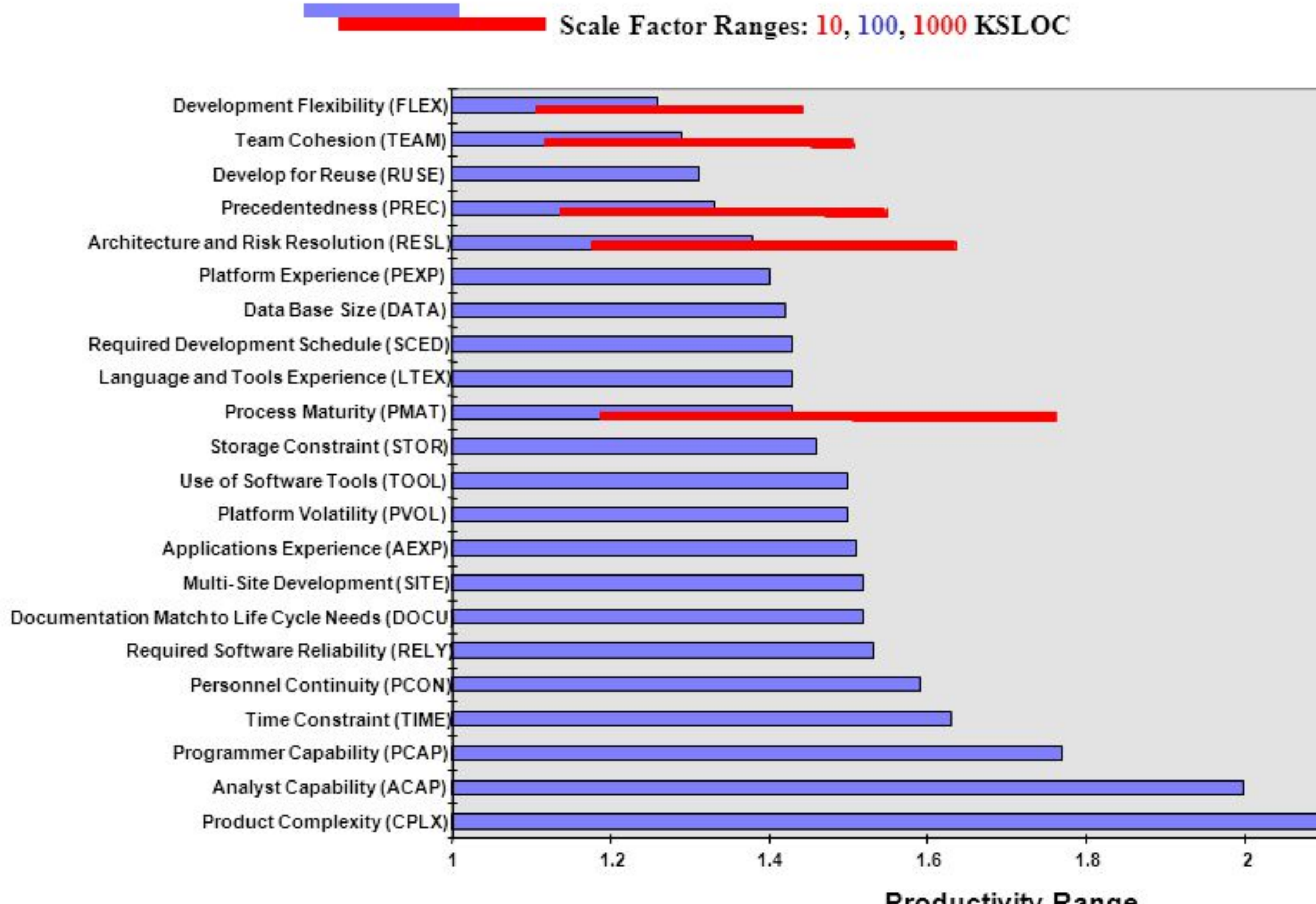
Estimate Influences

- SIZE OF THE SOFTWARE
  - *Most significant contributor to project effort and schedule*

Estimate Influences

- Effort scales up exponentially as Project Size does
- **Exponentially.** Not linearly.

COCOMO II. 2000 Productivity Rang



Story Points

- A mark of complexity
- Do NOT represent time!
  - *3 point story will take junior engineer lots of time, but senior engineer far lesser time*
  - *But… both will think of complexity in similar fashion*

Story Points

- Not complex…
  - *Create a lint step on the CI server*
  - *Add “Patient has already visited” option to Overdue list*
  - *Setup redis to support Call Sessions and Async Jobs*

Story Points

- Complex?
  - *Allow BPs to be deleted…*

- [x] Add "Remove" button to BP edit sheet
- [x] Show confirmation dialog for removing the BP
- [x] Update the `deletedAt` property for the `BloodPressureMeasurement`
- [x] Set the `syncStatus` on the `BloodPressureMeasurement` to `PENDING`
- [x] Close Edit BP sheet if "Remove" was clicked when confirming delete
- [x] Exclude deleted BPs when fetching in Patient Summary
- [x] Exclude deleted BPs when searching for patients (v2)
- [x] Exclude deleted BPs when sorting by facility in search results

Story Points

- Complex?
  - *Sync “Protocol Drugs” across facilities and display in app; If sync fails, show a list of default medicines*

- [x] Create `ProtocolRepository` that implements `SynceableRepository` using `ProtocolWithDrugs` that implements the `SynceableRepository`
- [x] Add protocol sync last pull process token to preferences
- [x] Add Retrofit service for protocol sync
- [x] Add `ProtocolSync` class to sync protocol and drugs
- [x] Add `ProtocolSync` to the `SyncWorker`. This should implement `ModelSync` and an instance of `ProtocolSync` must be added to the `ModelSyncTest` class
- [x] Add `ProtocolSyncCoordinatorAndroidTest` that extends `BaseSyncCoordinatorAndroidTest` (See `FacilitySyncAndroidTest` for certain limitations)
- [x] Create `ProtocolModule` to provide `ProtocolDao`

- [x] Create new package `org/simple/clinic/protocolv2`
- [x] Protocol Room model
- [x] Protocol drug Room model
- [x] Add migration for new protocol and protocol drug tables
- [x] Protocol Dao with method to save protocol
- [x] Protocol Drug Dao with method to save drugs, get drugs by protocol ID
- [x] Protocol payload model
- [x] Protocol drug payload model
- [x] Create combined model (`ProtocolWithDrugs`) for `Protocol` and `ProtocolDrug` to be used as the generic type for `ProtocolRepository`. See `PatientRepository` and `PatientProfile` for a working example.

Law of Large Numbers

- If you create One Big Estimate, your error tendency will either be completely on the high side or the low side

Law of Large Numbers

- But if you create multiple Small Estimates, some will be on the high side, some will be on the low side
- Errors will cancel out each other to some degree

Story Points

- Start with categorising stories by complexity
- After some iterations, velocity will tie in to time for the team
  - *“X team does 23 points in 1 iteration”*
  - *“Y team does 9 points in 1 iteration”*
- It takes some iterations to settle on a number

Task Estimates

- People who will actually DO the work should create granular estimates
- Ranges are important for realistic estimates

Task Estimates

- Developers usually do single-point estimates:
  - *Feature 1 — 2p*
  - *Feature 2 — 1p*
  - *Feature 3 — 3p*
  - *Feature 4 — 1p*

Task Estimates

- Follow up SPE by Best Case Estimates and Worst Case Estimates
  - *Feature 1 — 2p :: 1.5p | 3p*
  - *Feature 2 — 1p :: 1p | 2p*
  - *Feature 3 — 3p :: 2p | 5p*
  - *Feature 4 — 1p :: 0.5p | 2p*

Task Estimates

- "What happens if kinda sorta everything goes wrong?"
- Thinking about the WCE sometimes exposes additional work that must be done even in the BCE

Task Estimates

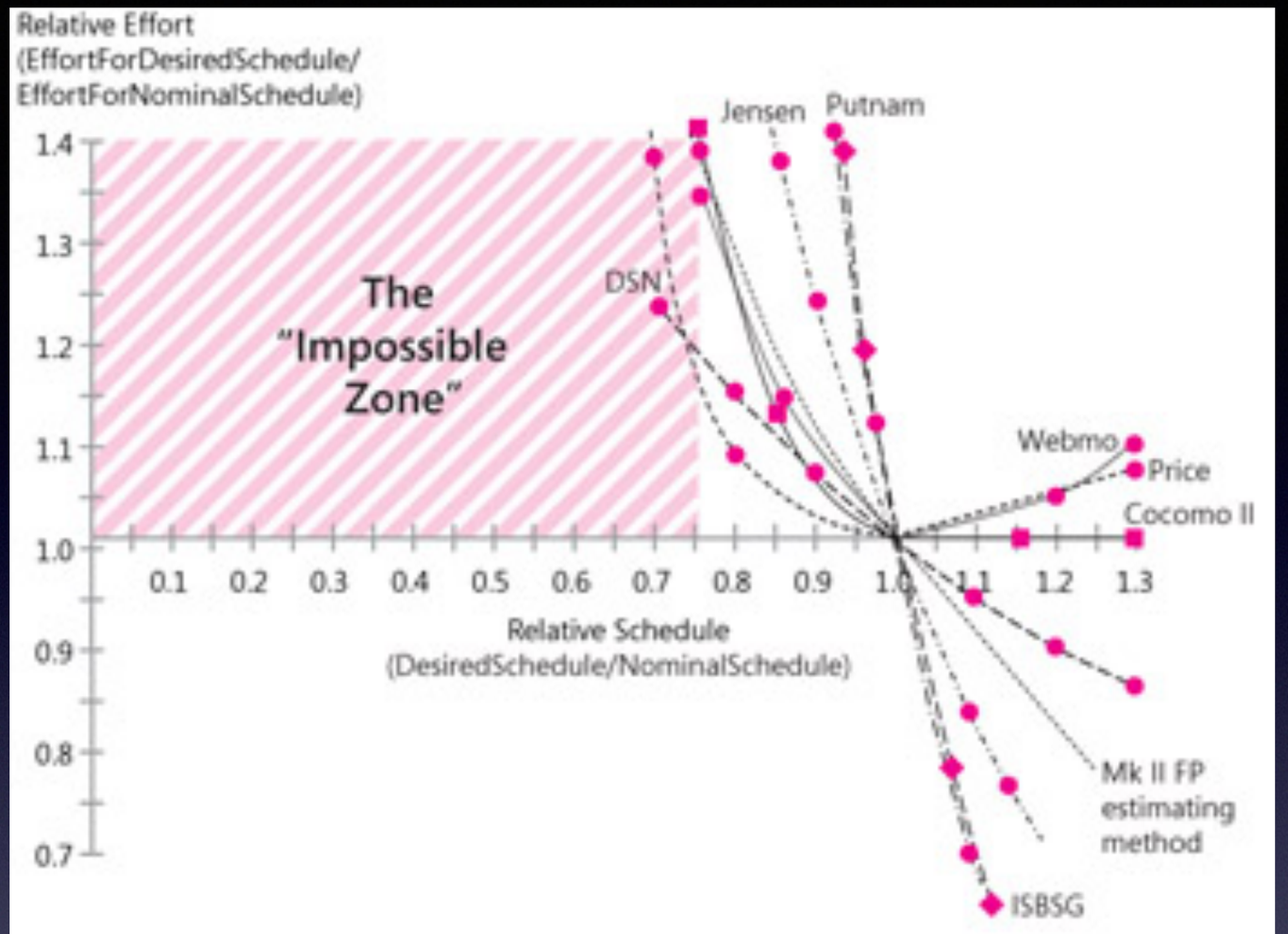
- Try comparing completely different stories which have same estimate
  - *Pick 4-5 random 1pt, 2pt, etc. stories*
  - *They should roughly be the same amount of effort!*

When shit goes down

- You have a 14 week schedule. First milestone was in 4 weeks but it took you 6 weeks. Should you…
  - *Make up 2 lost weeks later in the schedule?*
  - *Add 2 weeks to the total schedule?*
  - *Multiply the whole schedule by the magnitude of slip, i.e. 1.5x?*

Re-estimate!

- Do it multiple times during the project
- Collect historical data so you can predict with some confidence



Source: Adapted and extended from Software Sizing and Estimating: Mk II *(Symons 1991)*, Software Cost Estimation with Cocomo II *(Boehm et al 2000)*, *"Estimating Web Development Costs: There Are Differences" (Reifer 2002), and* Practical Project Estimation, 2nd Edition *(ISBSG 2005)*.

The Impossible Zone

The Impossible Zone

- Several decades of work
- Numerous researchers
- Many fucktons of data

The Impossible Zone

- Shortening the schedule increases effort:
  - *Large teams require more coordination and communication*
  - *More work needs to be done in parallel*
  - *Effort exponentially goes up*
  - *Burnout, stress, madness*

The Impossible Zone

- Extending the schedule usually reduces effort but **only if you reduce team size!**
- Schedule cannot be less than…

```
2 * ( StaffMonths ^ 0.34 )
```

Breaking the Impasse

- Non-technical stakeholders want (and need) to make decisions about scope
- But… good estimates take a lot of time and practice to do

Breaking the Impasse

- **Important!** The goal of estimation is not pinpoint accuracy... but accurate enough to support project control
- People don't care about number of days, they care about value in comparison to effort

Breaking the Impasse

- Do t-shirt sizing with the non-technical stakeholders!

Negotiation Emotions

- Bad Ideas:
  - *Strength of bargaining position, friendship*
  - *Desire to gain approval*
  - *Deception, intimidation*

Negotiation Emotions

- Four Pointers:
  - *Separate People from the Problem*
  - *Focus on Interests, not Positions*
  - *Invent options for Mutual Gains*
  - *Insist on Objective Criteria*

Negotiation Emotions

- Technical Staff owns the **Estimate**
- Non-technical Staff owns the **Target**

Negotiation Emotions

- Executives are usually assertive by job description
- Be aware of external influences. Communicate that you understand business requirements and their importance
- Negotiate the commitment, not the estimate

Negotiation Emotions

- Estimations are problems to be solved, not negotiated
- Recognise that all project stakeholders are on the same side
- Either everyone wins or everyone loses

Data to Collect

- Size: lines of code
- Effort: staff months/weeks
- Time: calendar months
- Defects: classified by severity

A Good Line

- "Clearly and obviously, half the programmers in the industry are below average, but I rarely meet project managers or executives who believe *their* people are the ones below average."