

Project planning

01219245/01219246
Individual Software Process

4 iterations

- 1st iteration: now – 26th March
 - Iteration review: 26th March
- 2nd iteration: 27th March – 9th April
 - Mid iteration review: 2nd April
 - Iteration review: 9th April
- 3rd iteration: 10th April – 23rd April
 - Mid iteration review: 16th April (*optional.*)
 - Iteration review: 23rd April
- Final iteration: 24th April – 7th May
 - Mid iteration review: 30th April
 - Project review: 7th May (hopefully with a project fair)

Calendar

Grey	Grey	Grey	Grey	Grey	Grey	Red
Red	White	White	White	White	T	Red
Red	White	White	White	White	White	Red
Green	Green	Green	Green	Green	Green	Green
Red	White	White	Grey	White	White	Red
Red	White	Grey	Grey	Grey	Grey	Grey

March

Grey	Grey	White	1	White	White	Red
Red	White	White	m	White	White	Red
Red	White	White	2	White	White	Red
Red	White	White	m	White	White	Red
Red	White	White	3	Grey	Grey	Grey

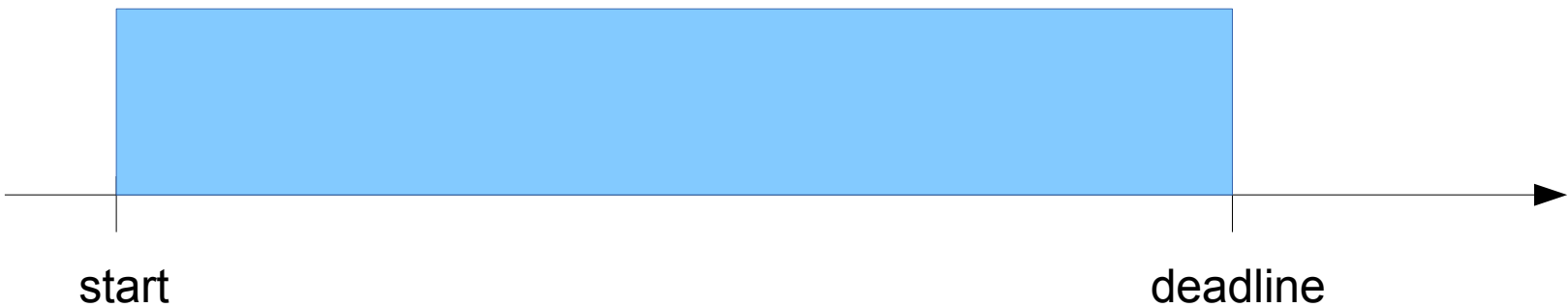
April

Grey	Grey	Grey	Grey	White	White	Red
Red	White	White	m	White	White	Red
Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Blue
Blue	Blue	Blue	Blue	Blue	Blue	Blue

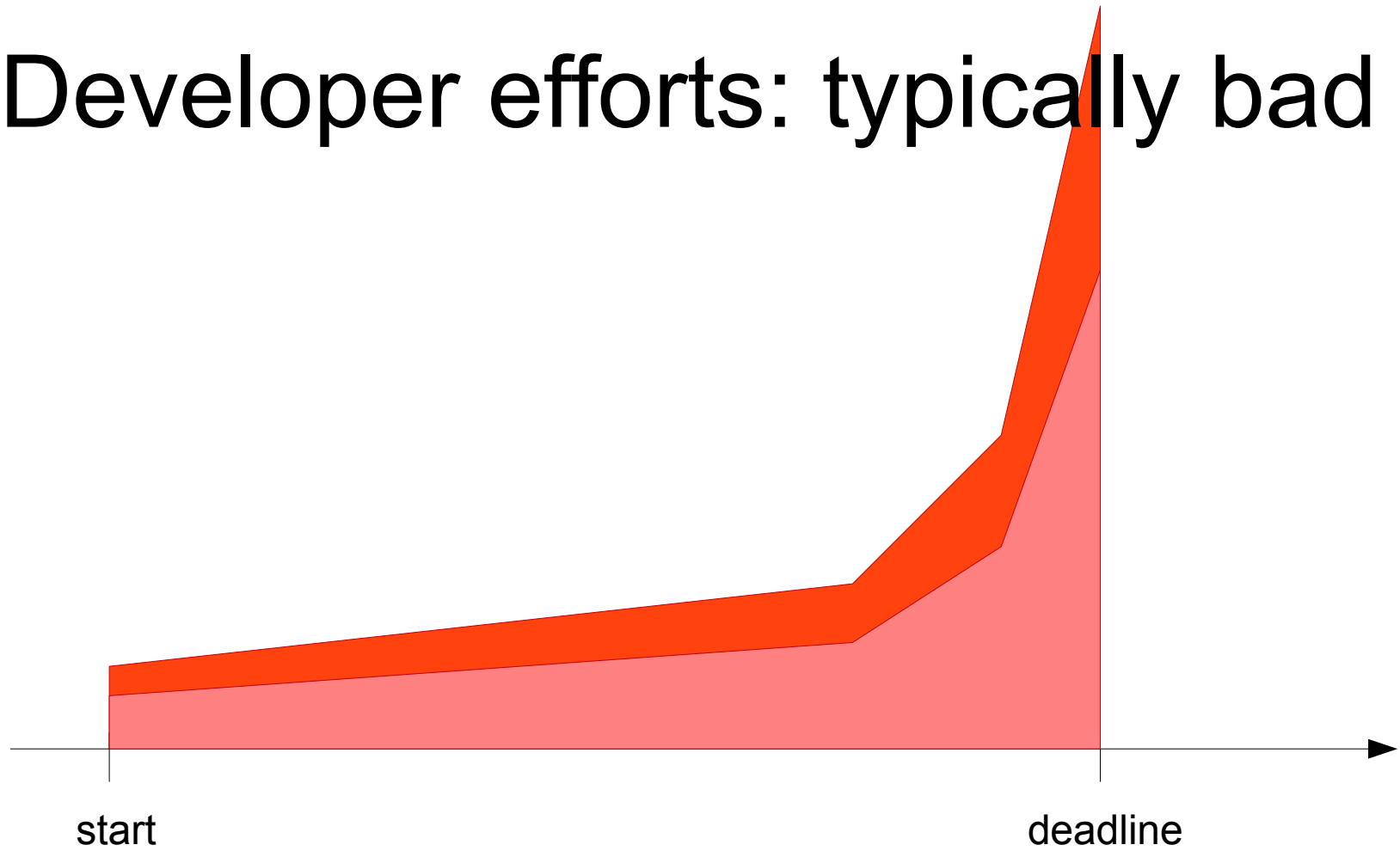
May

Planning

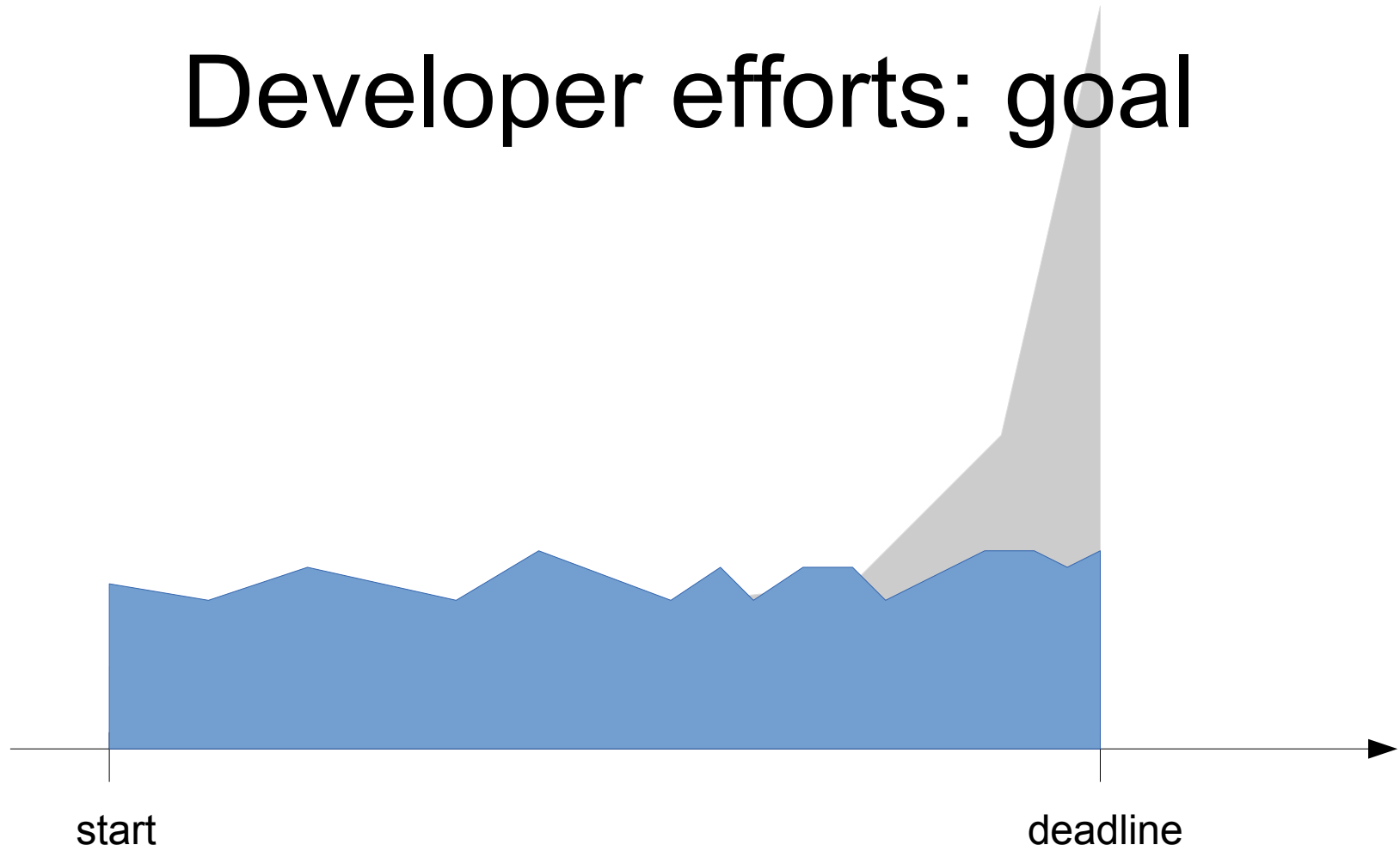
Developer efforts: ideal



Developer efforts: typically bad



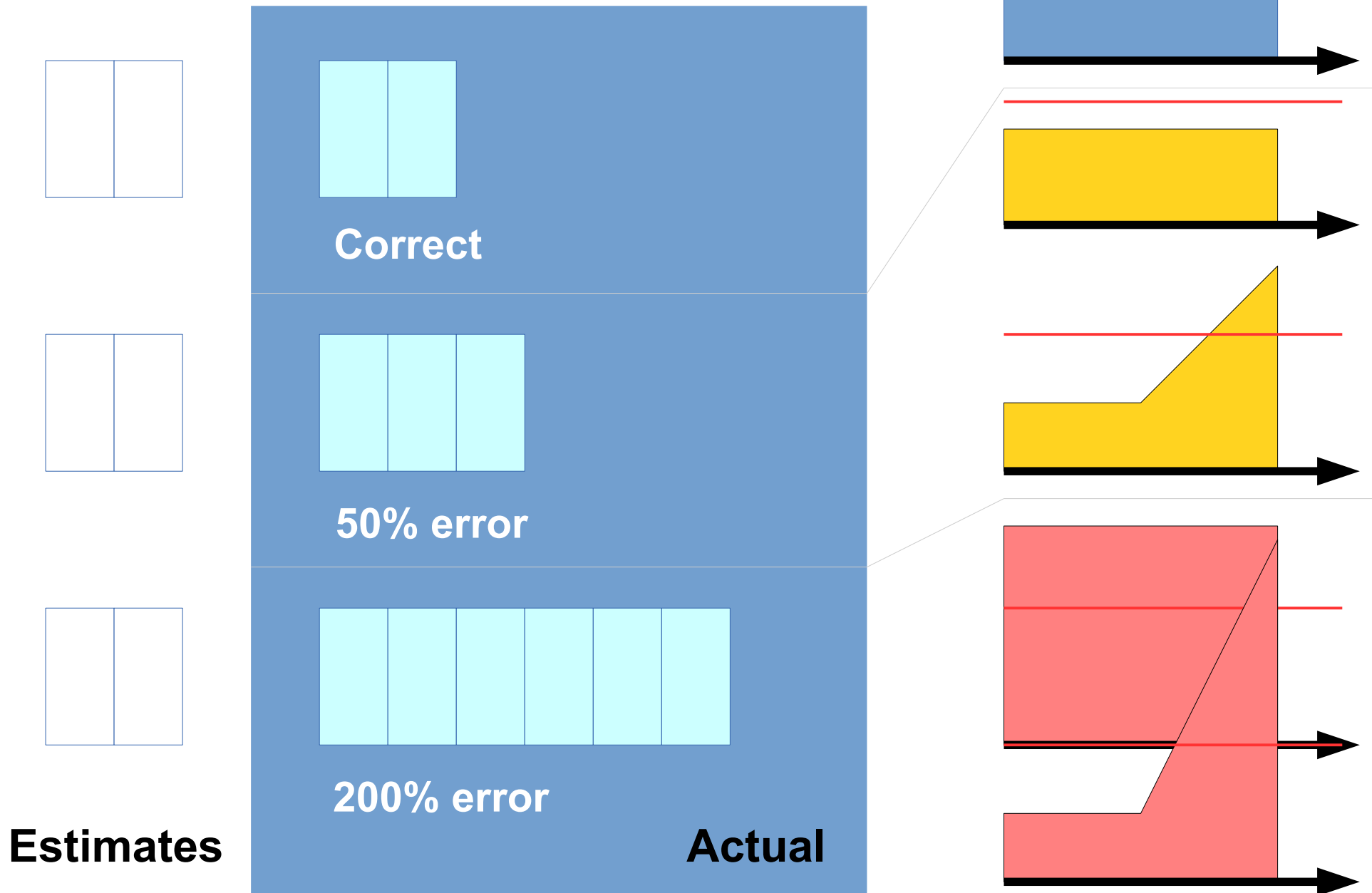
Developer efforts: goal



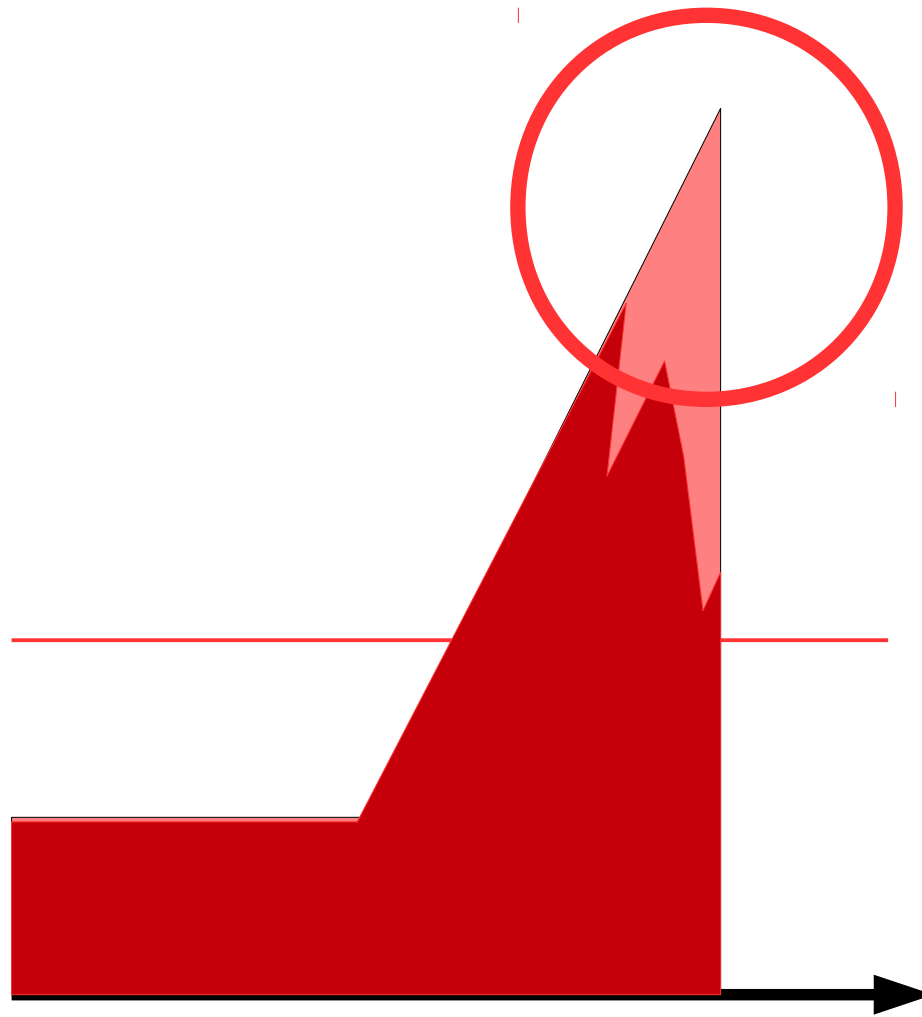
Why work so hard?

- Laziness? Not really.
- Usually, we underestimate the amount of work needed.

Sad story

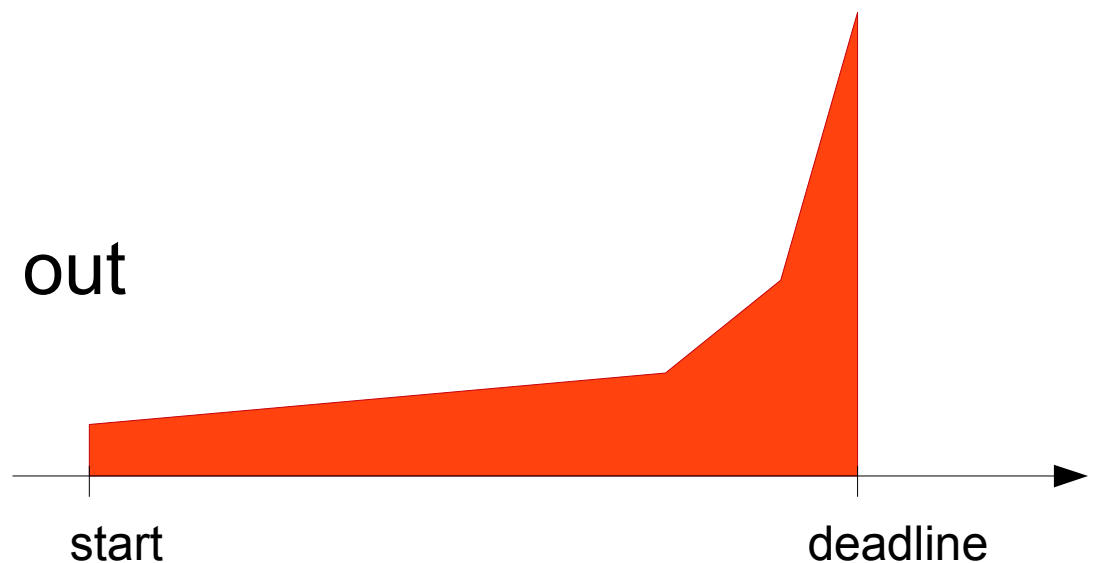


Can you do this?



Bad estimates

- We want to get all features done in a given time.
- However, we usually underestimate the effort needed.
- We end up with having all features but...
 - with bugs
 - with bad quality
 - and developers burn out



Good estimates

- If you have a good estimate, you can have a very realizable plan.
- But it is hard to have good estimates on the effort you need to get your software done.
- Very hard to obtain good estimates:
 - when you are doing something you have not done before,
 - when your customer's requirements are not clear,
 - when situations can change over time.

Dilemma?

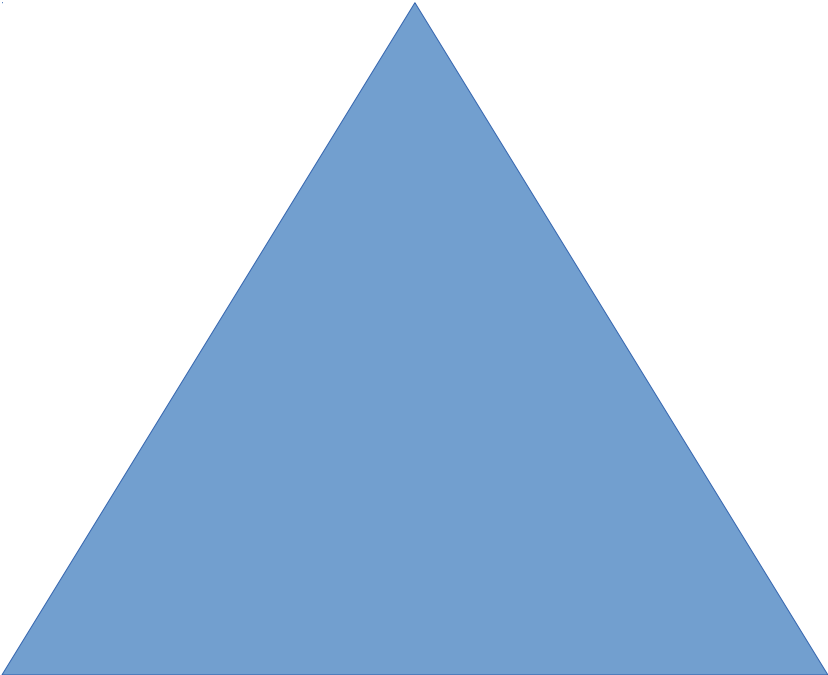
- 2 choices:
 - Find a way to obtain better estimates on the effort so that we can plan better.
 - Try to live without good estimates.
- Why can't we try both?

Trade-offs

Features

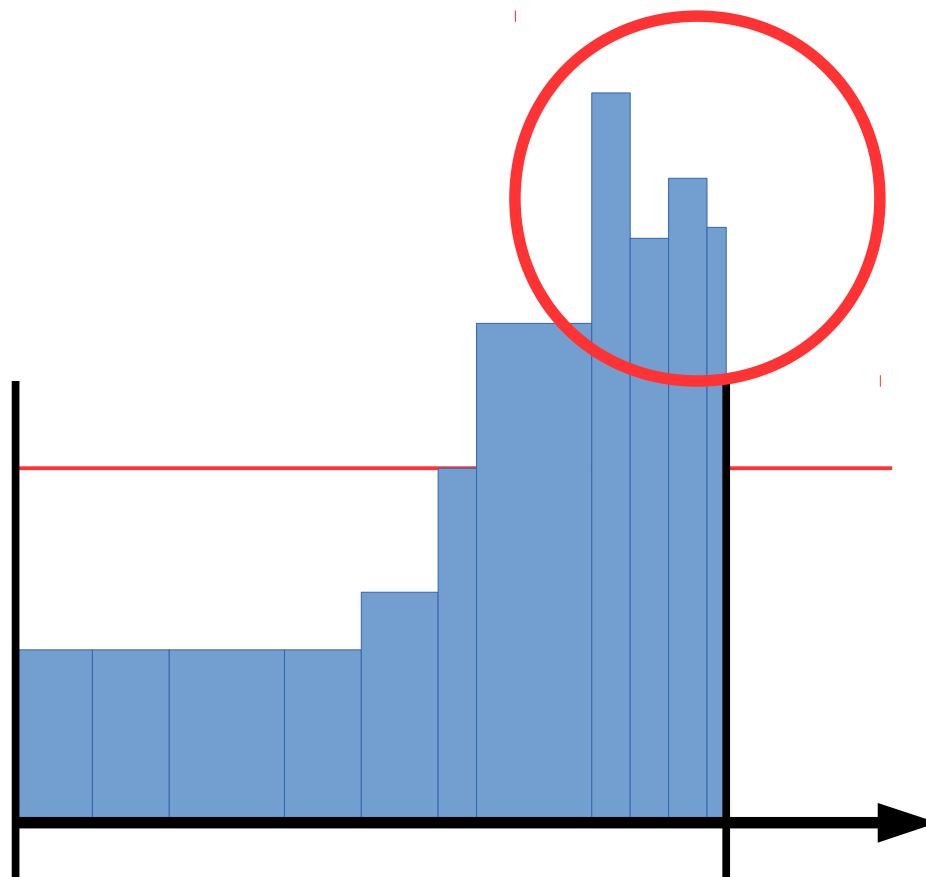
Time

Quality



Traditional trade-off

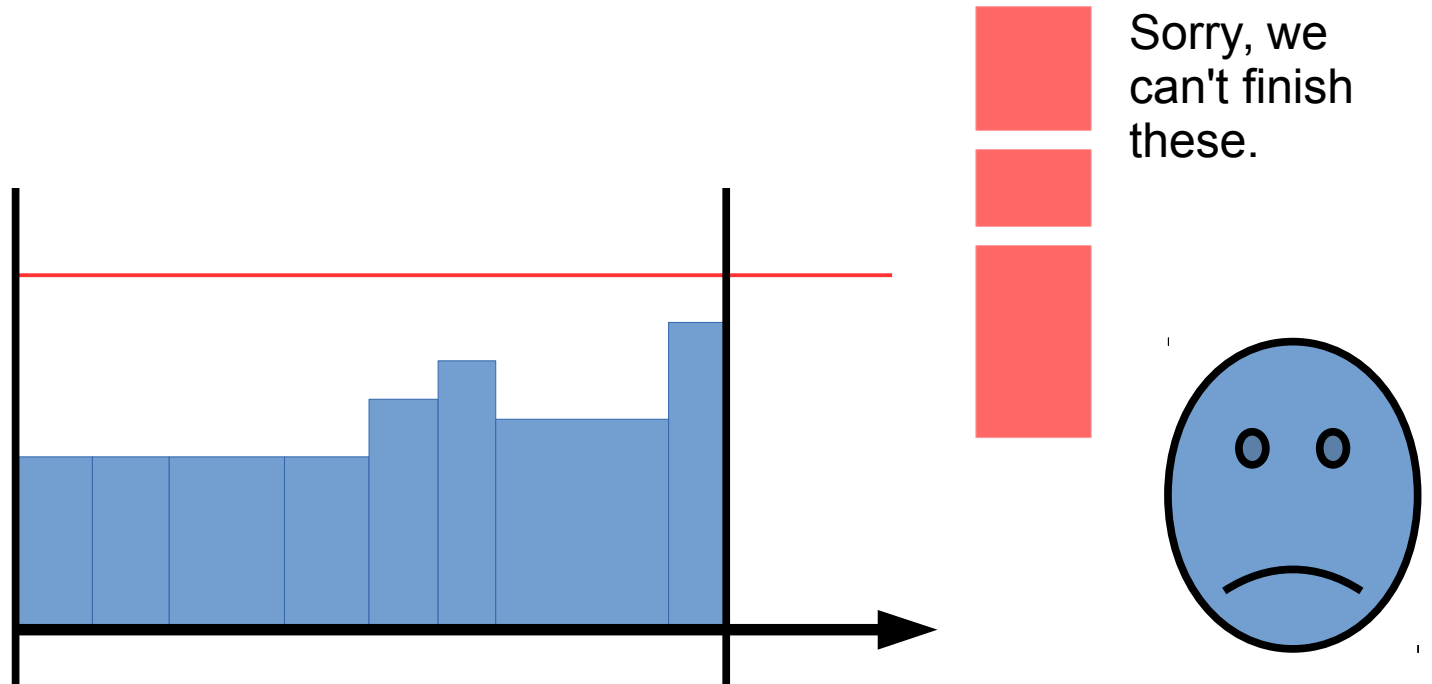
- Fixed time and features



Are we sure
that these features
are “done”?

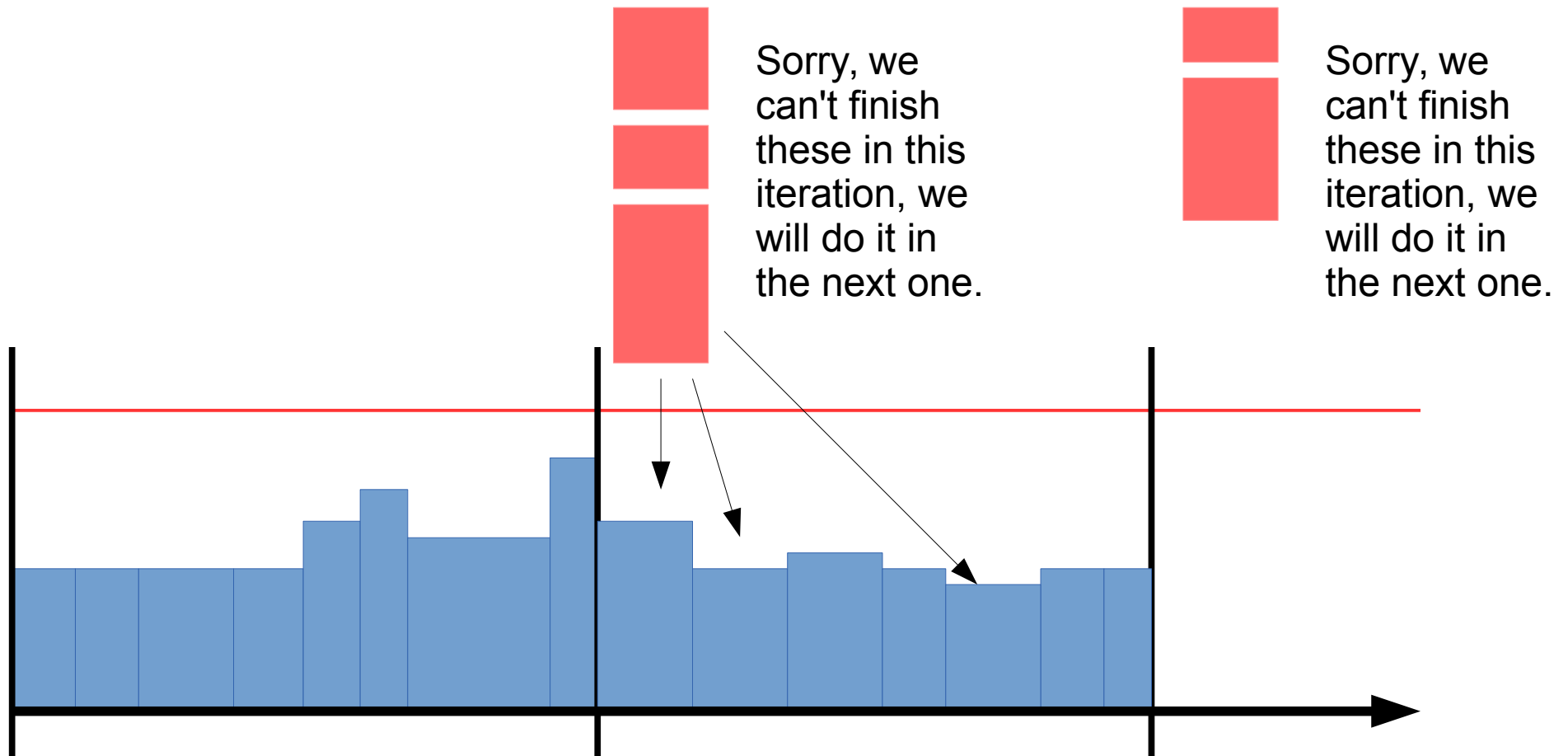
Live-and-learn trade-off

- Fixed time and quality



Live-and-learn trade-off with short iterations

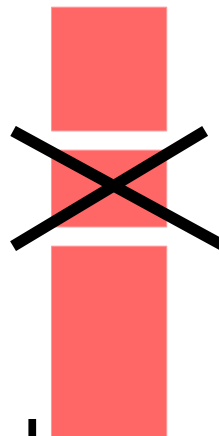
- Fixed time and quality



Live-and-learn trade-off with short iterations + re-planning

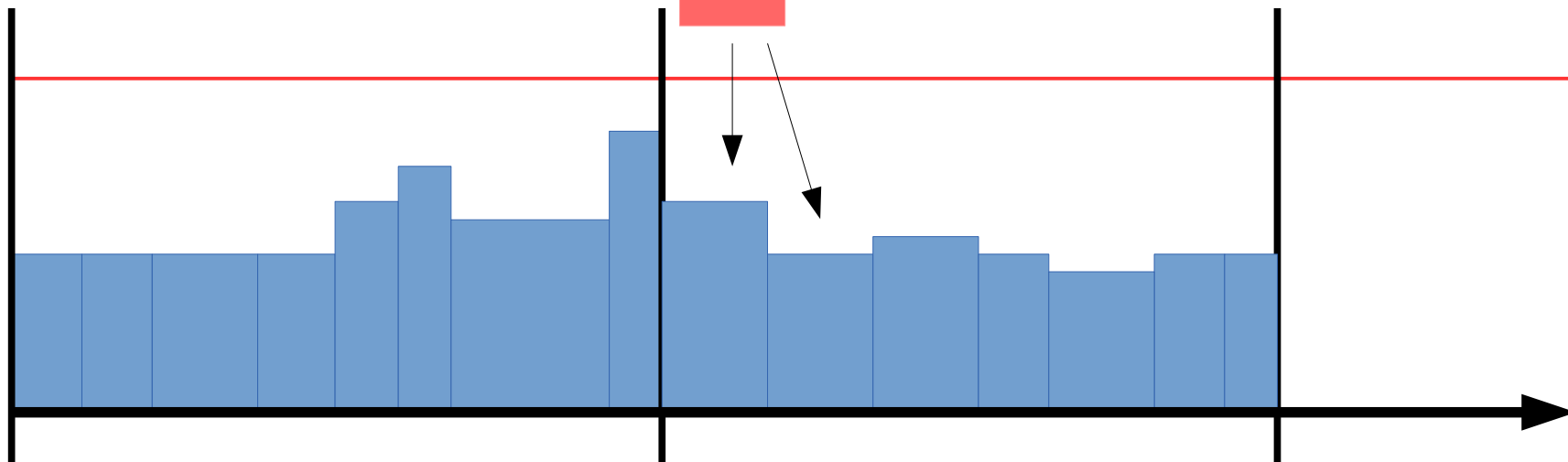
- Fixed time and quality

Sorry, we can't finish these in this iteration, we will do it in the next one.



OK. By the way, situation changes; you don't have to do this.

Sorry, we can't finish these in this iteration, we will do it in the next one.



Activities in planning

- Task breakdown
 - Smaller tasks are typically easier to estimate and to get done.
- Prioritization
 - We can't get everything done in one iteration. We should try to focus on more important ones first.
- Estimation
 - So that we have some idea on what features we can complete in this iteration.

Planning for your 1st Iteration

- No “formal” estimation at this point
 - since we really have no data.
- Set plausible goals for your 1st iteration
 - these are the features that we will complete first.
 - (see more on the next slides)
- Refine your task breakdown to correspond to the goals