D10 UNIT 4-B
RISK MANAGEMENT

Uncertainty, mitigation and contingency

Uncertainty

Uncertainty sources

- *Requirements:* What exactly does the system have to do?
- *Match:* How will the system interact with its human operators and other peer systems?
- *Changing environment:* How will needs and goals change during the period of development?
- *Resources:* What key human skills will be available as the project proceeds?

Uncertainty
Uncertainty sources

- *Management:* Will management set up productive teams, maintain morale, keep turnover low, and coordinate complex sets of interrelated tasks?
- *Supply chain:* Will other parties to the development perform as hoped?
- *Politics:* What is the effect of political decisions on the project?

Uncertainty

Uncertainty sources

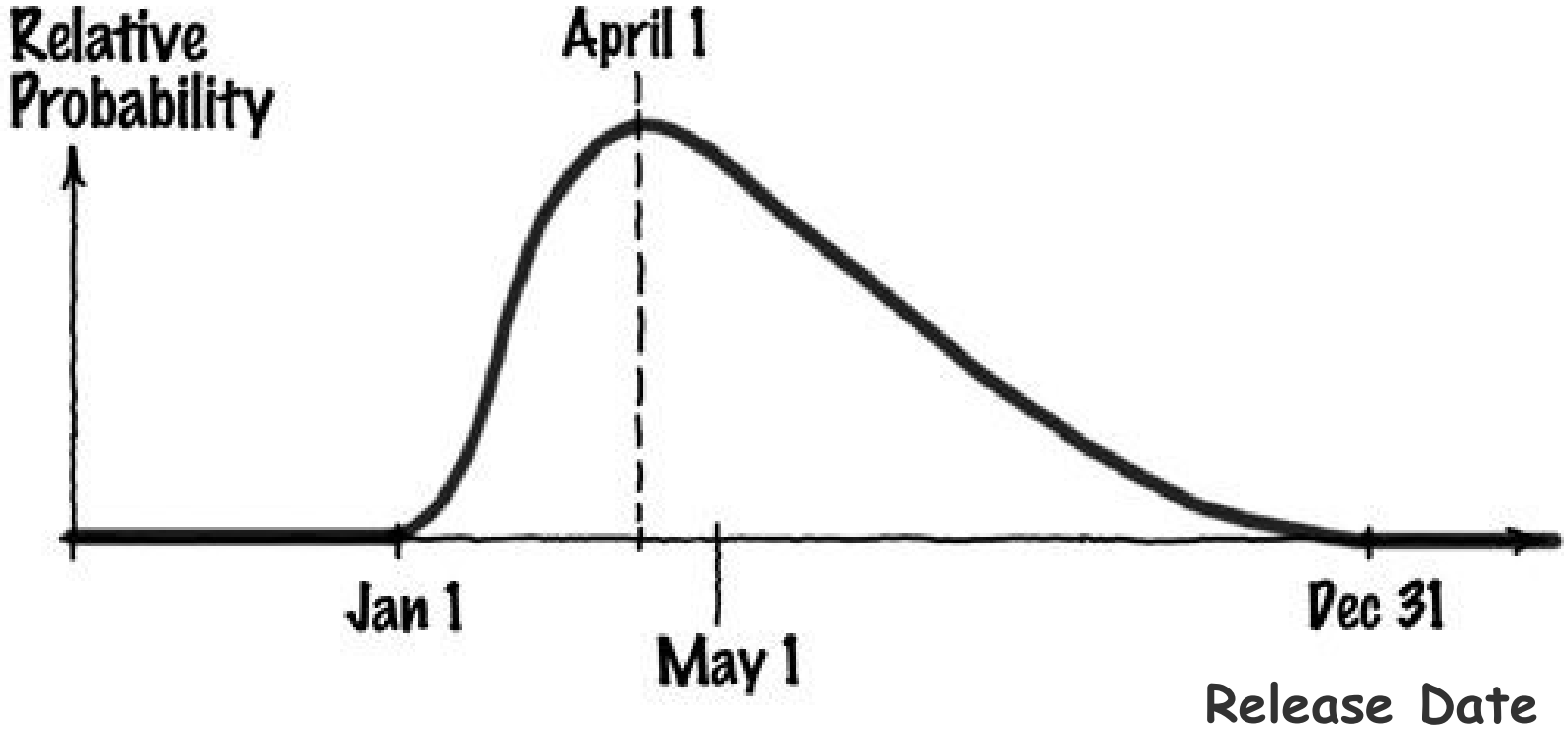
- *Conflict:* How do members of a diverse stakeholder community resolve their mutually incompatible goals?
- *Innovation:* How will technologies and approaches affect the eventual outcome?
- *Scale:* How will upscaling volume and scope beyond past experience impact project performance?

Uncertainty

Uncertainty in software projects

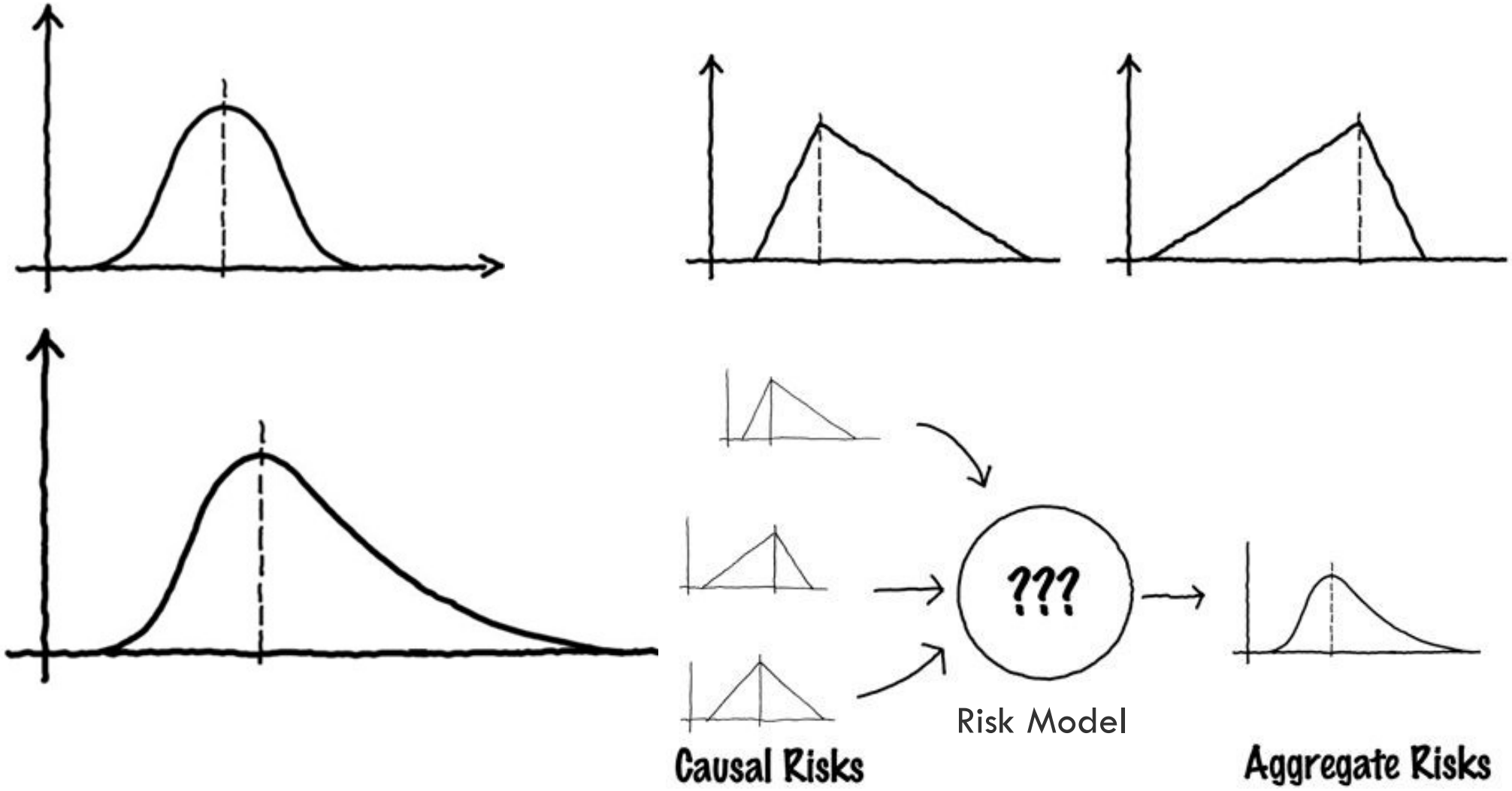
Most software project managers do a reasonable job of predicting the tasks that *have to be done* and a poor job of predicting the tasks that *might have to be done.*

Uncertainty
Quantifying uncertainty



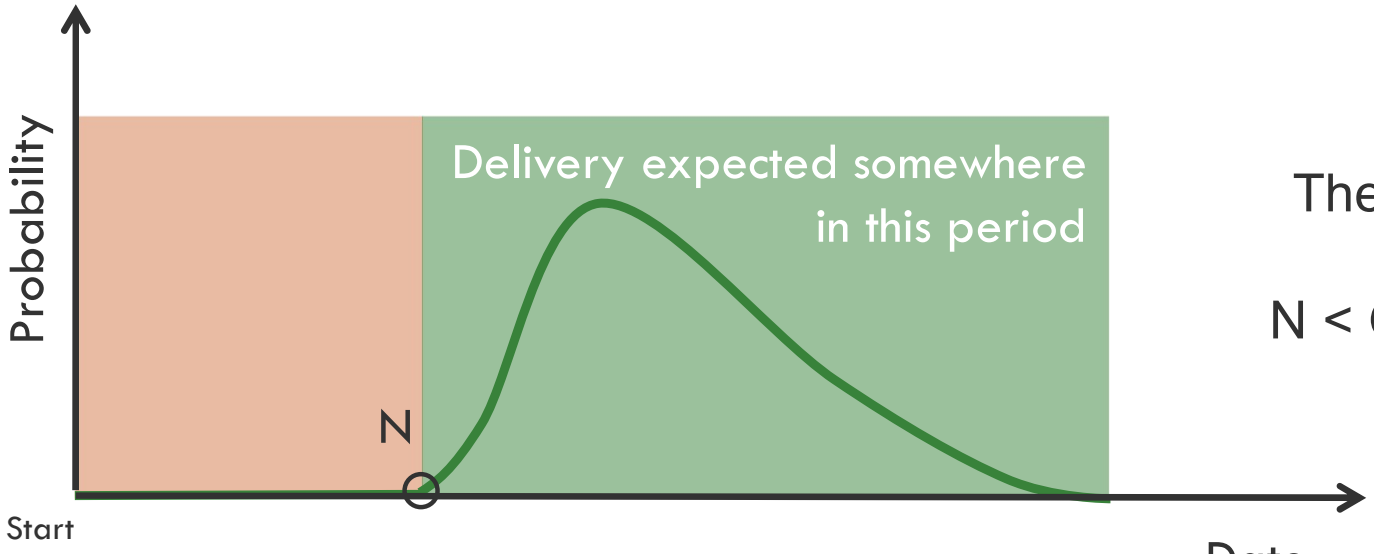
Risk diagram (release date uncertainty)

Uncertainty
Quantifying uncertainty



Uncertainty
Quantifying uncertainty

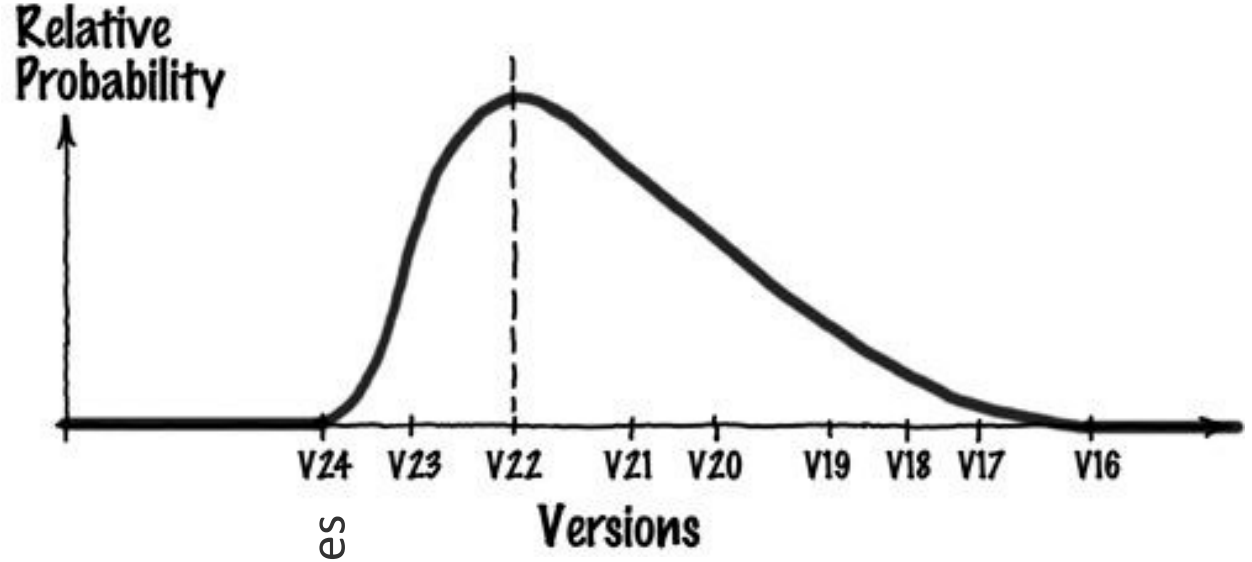
- In general, uncertainty windows of 10 to 15 percent of N (nano-percent date) are proper.
- For the software industry as a whole, window size is in the range of 150 to 200 percent of N.
- The size of your window of uncertainty is a function of how much noise (variation) there is in your organization's development processes
- Your past performance determines window size



The most sensible:

N < Goal < Schedule

Uncertainty
Functional uncertainty



All functionalities

Some functionalities

Risk diagram (functional uncertainty)

The date is now fixed and the uncertainty is entirely a matter of what will be delivered on that date.

What to do about a risk

Alternatives

- Avoid 
- Contain 
- Mitigate 
- Evade 

What to do about a risk

Avoid

You *avoid* a risk when you don’t do the project or the part of the project that entails the risk. The natural consequence of avoiding a risk is that you forgo the benefit that going into the risky area offered.

What to do about a risk

Contain

You *contain* a risk when you set aside sufficient time and money to pay for it, should it materialize.

In practice, it doesn't make much sense to contain a single risk; instead, you contain your entire set of risks. Some of them will materialize and others won't.

A containment strategy sets aside enough resources, on average, to offset the risks that are likely to materialize.

What to do about a risk

Mitigate

You *mitigate* a risk when you take steps before its materialization to reduce eventual containment costs or its probability.

This includes the steps required in advance so that the containment strategy you've chosen will be implementable at transition time.

What to do about a risk

Evade

You *evade* a risk when you do none of the other options and the risk just doesn't materialize. Fortunately, some risks *expire* during the course of a project.

What to do about a risk

Cost

The first three of these cost money:

- Avoidance costs you lost benefit.
- Containment costs you the portion of risk reserves that gets used up.
- Mitigation costs you whatever you spend to reduce containment cost or probability.
- Only risk evasion appears to be free.

What to do about a risk

Mitigation vs Contingency

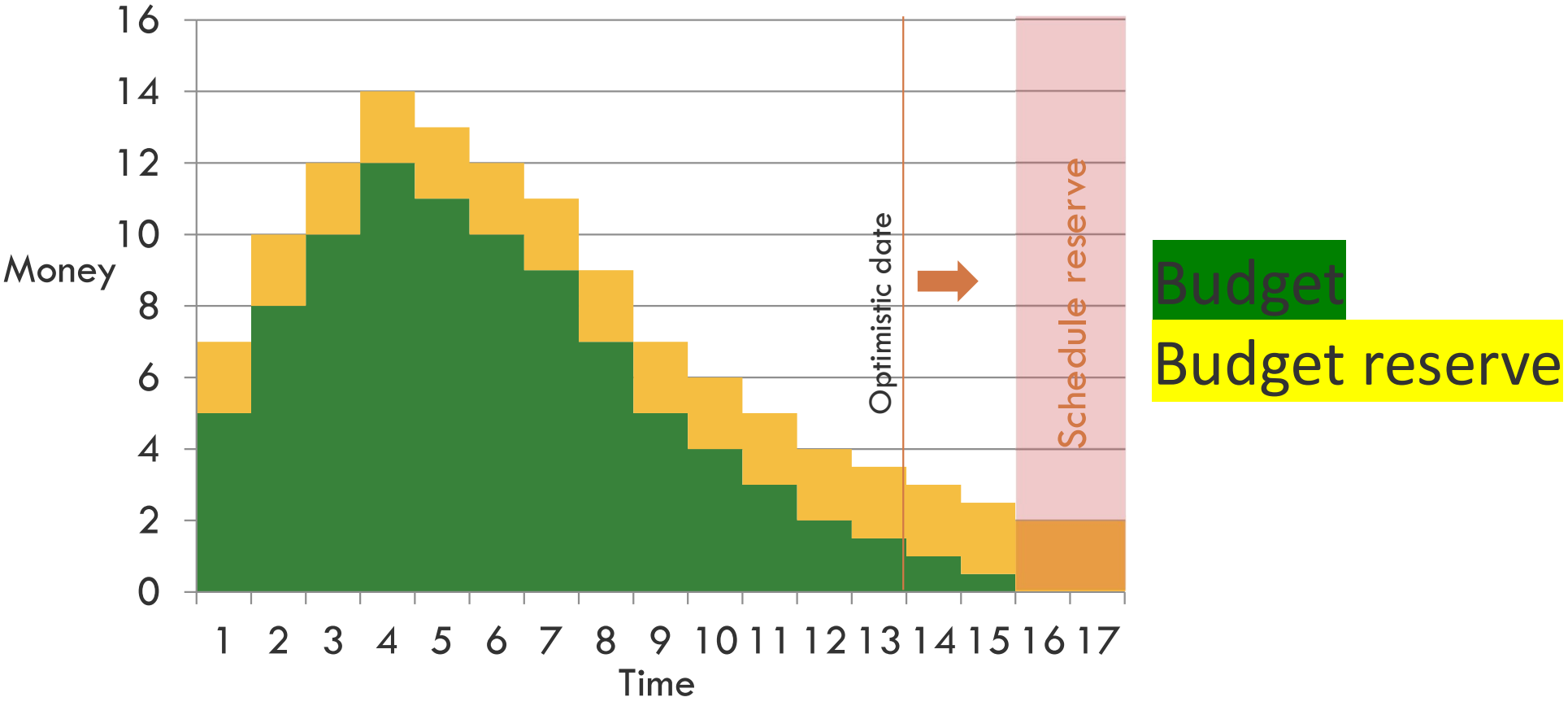
Risk mitigation plan	Risk contingency plan
You identify actions which you will take in advance irrespective of the occurrence of risk.	You plan actions, but you monitor certain warning signs. You take these actions only when you see the warning signs.
You spend time and money in advance for the given risk condition.	You do not spend time or money in advance, but you keep them ready and invest them when needed.
We are expected to mitigate the risks which are outside the risk threshold. By applying a mitigation plan, we reduce the probability or the impact of the identified risk.	By identifying the contingency plan, we do not change the probability or impact of the current risk, but we plan to control the impact as risk event looks like occurring.
This works as the first level of defense for the high exposure risks	This works as a fallback plan for high exposure risks.

Risk reserve
Containment

- A risk reserve is a buffer of time and money set aside to contain risks.
- A risk reserve is, by definition, time and money that you may not need.
- If you calculate exposure for all your risks and set aside a risk reserve equal to the total exposure, that risk reserve will, on average, be sufficient to pay for the risks that do materialize. You may end up short on some projects and have some reserve left on others, but over the long run, your risk reserve will be about right.
- A more defensive strategy would be to allocate something more than aggregate exposure, while a less defensive strategy would be to allocate less.

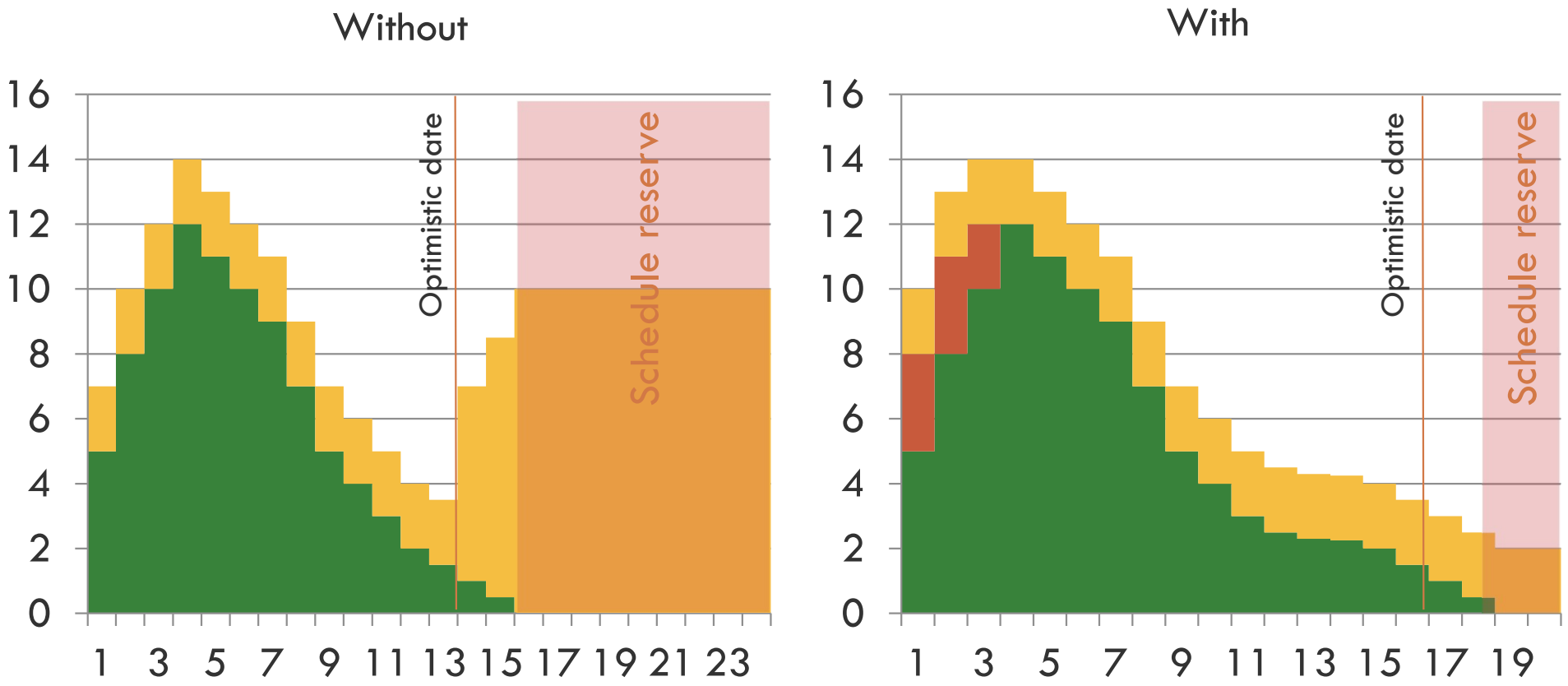
Risk reserve

Budget and schedule reserves



Risk reserve

Without and with mitigation



Risk reserve
Without and with mitigation

Without mitigation, the schedule reserve is large; The budget reserve can be enormous to pay for the additional cost.

With mitigation, both reserves are considerably reduced. However, the budget has been increased by the cost of mitigation. The schedule has also been stretched out somewhat to the right, since mitigation has time cost as well as an economic cost. The result is that the optimistic date on the graph is somewhat less optimistic than it was in the no-mitigation plan.

Transition monitoring

Transition indicators

- For each managed risk, you need to choose one or more early indications of materialization.
- The earliest indicator may expose you to false-positive signals.
- Your choice of transition indicator requires a thoughtful assessment of urgency and of the cost of false-positive triggering.

Examples
Risk mitigation

Risk 1: Estimation and scheduling
Explanation: Software development, given the intangible nature and uniqueness of software, is inherently difficult to estimate and schedule.
• Get the team more involved in planning and estimating. • Get early feedback and address slips directly with stakeholders. • By working in short increments the true velocity of the team quickly emerges and is visible to all stakeholders who are now more closely involved in the project. • Always monitor existing projects so that you apply lessons learnt in the future.

Examples
Risk mitigation

Risk 2: Requirements Inflation

Explanation: As the project progresses more and more features that were not identified at the beginning of the project emerge that threaten estimates and timelines.

- Constant involvement of customers and developers.
- Plan discussions about features and estimates at every iteration boundary.
- Rather than utilizing change-suppression mechanisms, prioritization sessions are scheduled that allow worthwhile changes to proceed and initially envisioned features to be superseded if the business gives their authorization.
- Try to think big early on in the project, and anticipate the worst-case or heaviest-use scenario.

Examples
Risk mitigation

Risk 3: Employee Turnover
Explanation: Key personnel leave the project taking critical information with them that significantly delays or derails the project.
• Increased collaboration and information sharing on the team. • Practice information sharing techniques such as pair programming, common code ownership, and frequent reporting at daily stand-ups specifically to reduce the "bus-factor". Multiple team members share key information and the risk due to employee turnover is small. • When working in an engaging, rewarding, empowered, collaborative environment such as agile projects, people are far less likely to want to move elsewhere.

Examples
Risk mitigation

Risk 4: Specification Breakdown
Explanation: When coding and integration begin it becomes apparent that the specification is incomplete or contains conflicting requirements.
• Use a dedicated Product Manager to make critical trade off decisions. • Make use of an ambassador user, subject matter expert, or customer proxy to play the product manager role. Agile projects have some form of product owner role to ensure decisions are made in a timely fashion. • Joint Application Development (JAD) can be helpful to collect requirements, enhance user participation and improve the quality of specifications.

Examples
Risk mitigation

Risk 5: Poor Productivity
Explanation: Given long project timelines, the sense of urgency to work in earnest is often absent resulting to time lost in early project stages that can never be regained.
• Short iterations, right people on team, coaching and team development. • Parkinson's Law says that: "Work expands to fill the time available" and Student Syndrome: "Given a deadline, people tend to wait until the deadline is nearly here before starting work." By having short iterations, work is time-boxed into a manageable iteration (typically 1-4 weeks) and there is always a sense of urgency. • Set a realistic schedule, and stick to it.

Examples
Risk mitigation

Risk 6: Budget

Explanation: Projects exceed their budget. Goals may change, assumptions fail, and changes may be necessary to make the Project viable. Very frequent, sometimes related to other risks.

- In product development, you necessarily make assumptions that cannot be proven or disproven until more information becomes available. As development progresses, objectives or goals may shift, or the product may need to pivot to be viable.
- Rolling wave planning is designed to account for this. Teams make product decisions when they are in the best position to make them, rather than presenting very detailed plans at the beginning of the project. This mitigates budget risk because you do not have to waste time and resources for re-planning.

Examples
Risk mitigation

Risk 7: Personnel, Knowledge
Explanation: Loss or absence of some members of the team. Even if it is for a short period, can result in delays or errors. Additional resources and cost for training may be needed.
• Squad-based development: Squads are 10-12 person, co-located teams that plan together, share knowledge, complete code reviews and work together on a given project from beginning to end. They have a known maximum capacity and open flow of knowledge, which helps address both personnel and knowledge risk, as it eliminates knowledge silos and gives team members the ability to seamlessly take on tasks if somebody is absent or leaves the team.

Some other common risks

Compromising on designs

In order to start doing the next ‘real’ tasks, developers tend to rush the design-process. This is a waste of programming hours, as designing is the most critical part of software development, and having to revisit it later in the process risks dragging everything back a few steps to ensure things like UX remain at a high level.

Gold plating

Developers sometimes like to show off their skills by adding unnecessary features. Again, this is a waste of programming hours, which could be better spent on sticking to the timeline for development and getting the software completed.

Some other common risks

Compromising on functionalities

Sometimes software development firms reduce the functionality of the software to compensate for overruns pertaining to high budgets and scheduling. There is always a conflict between achieving maximum functionality of the software and peak performance. This is a particularly dangerous moment in terms of app security.

Procedural risks

Day-to-day operational activities might hamper the software development plan and procedure, due to improper process implementation, conflicting priorities, or a lack of clarity in responsibilities.

References

Waltzing with Bears: Managing Risk on Software Projects
by Tom Demarco, Timothy Lister
Addison Wesley Professional, 2013

https://www.izenbridge.com/blog/know-the-difference-between-mitigation-plan-and-contingency-plan/

https://www.itproportal.com/features/top-10-software-development-risks/

https://www.projectsmart.co.uk/risk-management/top-five-software-project-risks.php