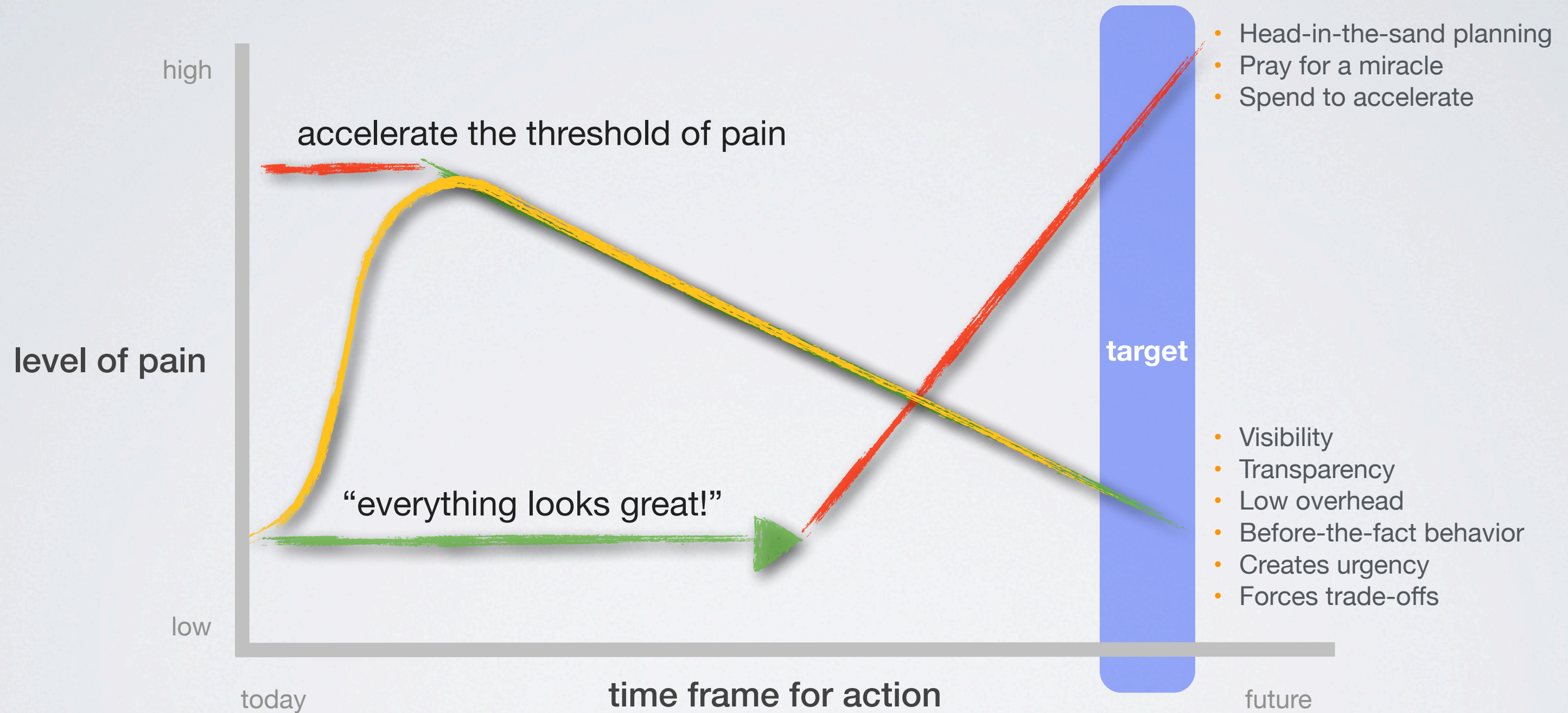


Concepts

Bring Pain Forward

Accelerate pain to create urgency and action today



Why do it?

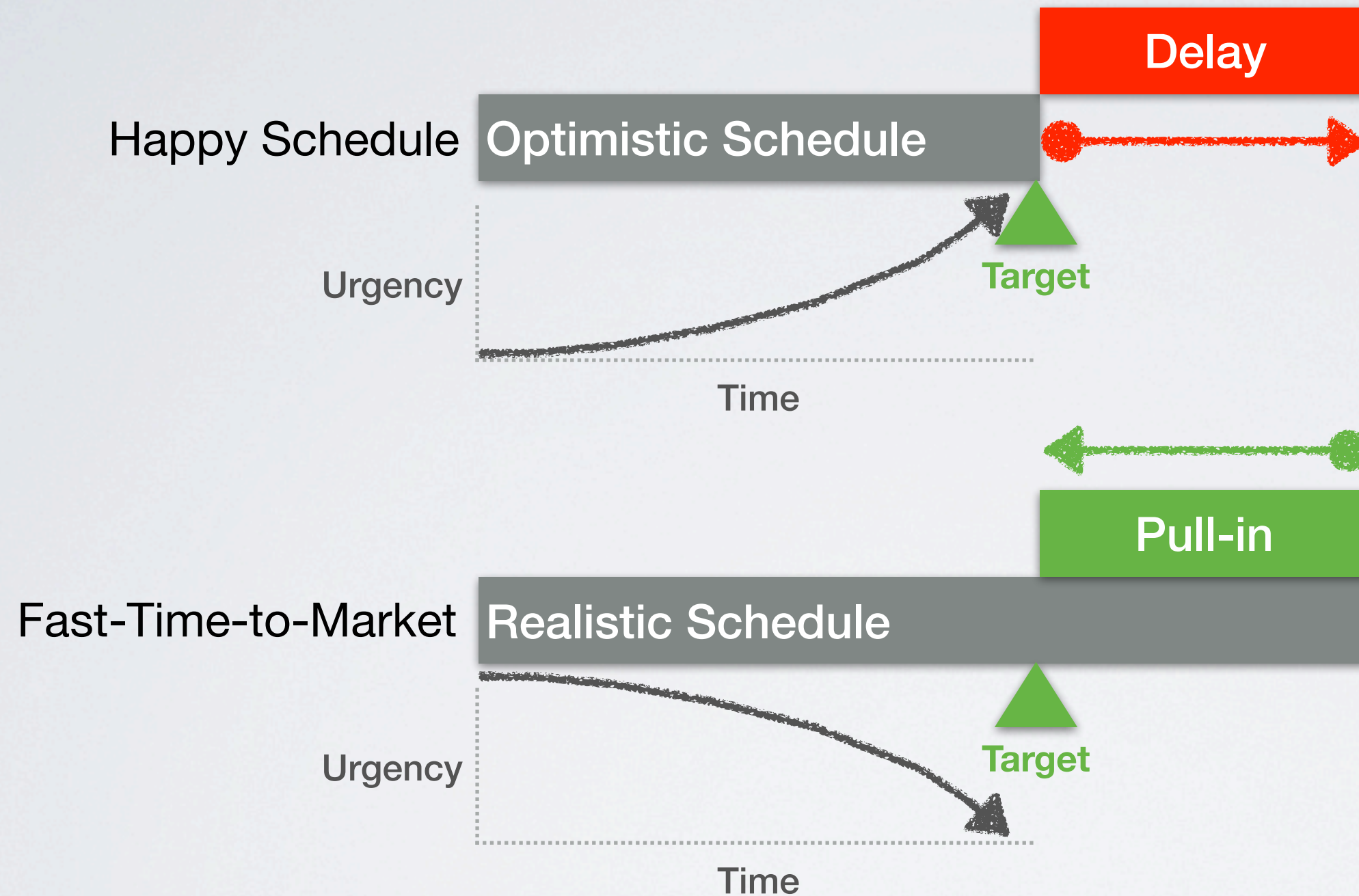
- To see future problems before they occur
- To create urgency, early vs later
- To address scope issues early
- To identify decisions today that could influence future outcomes
- To take action early, i.e. start early to give longer time to solve difficult problems
- Permits team to put in place mitigation actions, in advance
- Creates "before-the-fact" behavior across team

Why people don't do it?

- Don't want to see reality early; hope it will "all work out somehow" during the project
- Means one has to make difficult decisions, would rather push them ahead than make them now
- Permits team to pretend that they can do something that most of the members believe they can't (perpetuates the dream)
- The prolonged "dream state" makes people feel good
- Sometimes seen as management failure to identify failure in advance
- "After-the-fact" behavior is easier, since technical projects are complex, and complexity can always be blamed for the failure

Happy Schedules

Realistic Planning; identify gap



- “Nothing can go wrong” thinking
- Minimize risks
- Hope for “right first time” results
- Reactive and surprised
- Idealistic and top-down driven

- Factor in risk
- Factor in learning cycles
- Act before-the-fact to mitigate risks
- Proactive, anticipate
- Pragmatic and consensus based

Why do teams create happy schedules?

- Wishful thinking, hope they work out, but most know they won't
- Easier to avoid management punishment
- Supported by management culture, realistic schedules are discouraged because they can become "self fulfilling"
- Can't see the critical paths, so less pressure on individuals
- Permits teams to ignore known risks, in hope that these risks don't materialize (more wishful thinking)

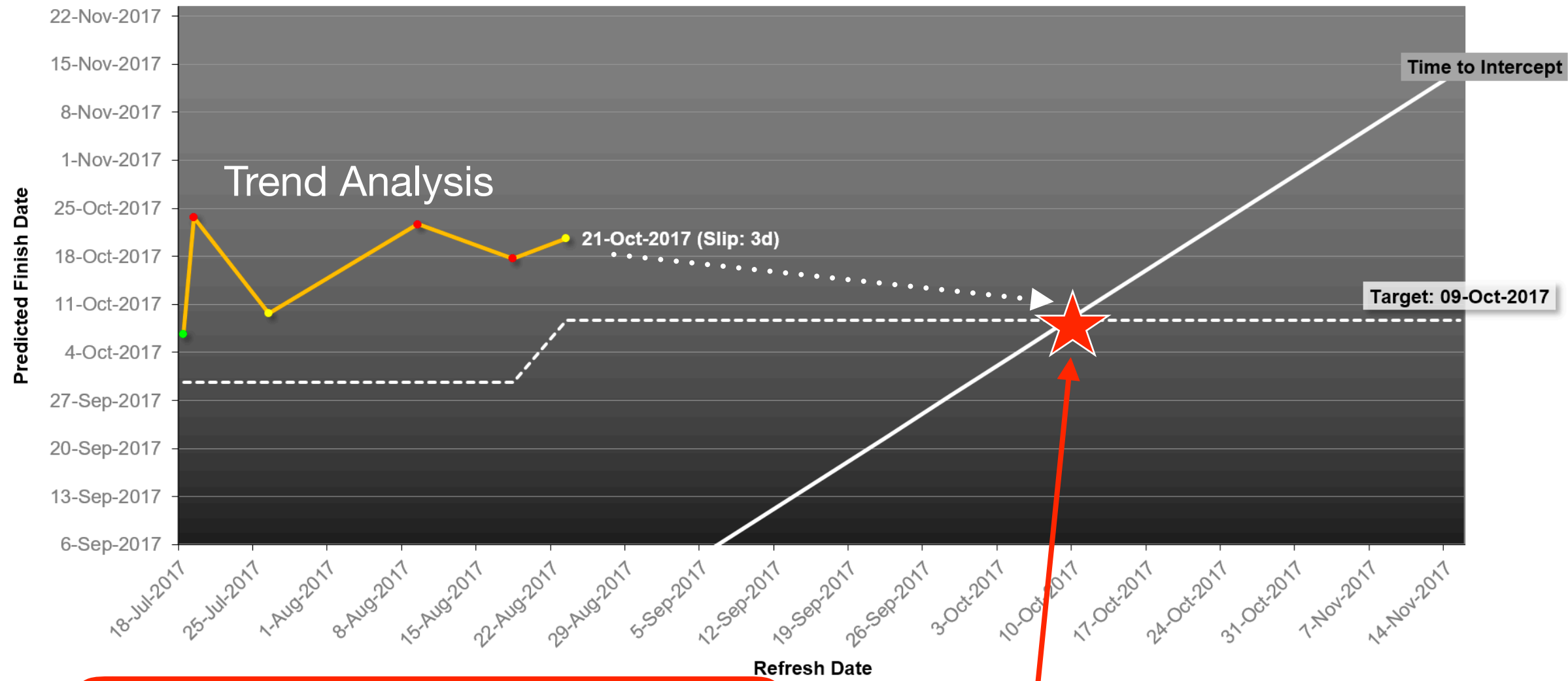
What is the advantage of a realistic schedule?

- Can see the critical paths, permitting focus and prioritization
- Prioritization means added resources to help along with extra support on the "difficult bits"
- Early warning of problems permit longer duration to solve them (start sooner, take longer)
- Permits proactive alignment between schedule & product requirements, rather than reactively redefining outcomes
- Risks are acknowledged and factored into planning to avoid future surprises
- When risks are factored into planning, they can be mitigated in advance
- Must be supported by a management environment that rewards early problem identification and honest behavior

Know the Gap

Status...

Target Milestone	Target	Schedule Date	Completed On	Gap (days)	Health
17M PPM achieved	9-Oct-2017	21-Oct-2017		Late: 12d	●



Done when...

- Done when:
- AFG NC Line 2, including semi-auto pack-out, fully qualified and ramped
- Rework implemented and ramped
- Defond CN __M PPM/week (2.25M/week * 90% derate), fully qualified and ramped
 - * Flextronics plastics
 - * ACE Plastics
- AFG SZ __M PPM, fully qualified and ramped (Machines 1-13)
- Facilities and waste disposal readiness?

Why do teams “hide” the gap?

- To avoid negative repercussions from management
- Contradicts the top down dictate to be done on x date; who wants to argue with the boss?
- To avoid being attacked for making bloated estimates, being overly “risk adverse”
- To avoid criticism for being the “negative non-team player” (“we need can-do, not can’t-do people)
- Prolongs the time needed to make difficult decisions - when delayed to the end of the project, they become “decisions by default,” so individuals don’t get blamed

Why to teams expose the gap, early?

- To force decisions early to align the scope with the timeframe
- To create urgency, because if the gap is known early, the team can generate energy around closing the gap
- Reality is sobering, better to be sober as soon as you can on challenging, bleeding edge projects
- To force breakthrough thinking (people get more creative when they are pressed against the wall)
- In order for this to work, the management environment must be set up to give individuals “permission” to be honest (fails when honesty is met with punishment or cynicism)

lateralworks >

CONFIDENTIAL AND PROPRIETARY

Any use of this material without specific permission of lateralworks is strictly prohibited.

© lateralworks. All rights reserved.

Common Problems

Multiple Sites

Characteristics

Team members distributed across many locations/time zones
Lack convenient regular meeting times
Disconnected and compartmentalized work flow
Lack continuity and ability of sub-teams to visualize and own the overall development project
Lack of overall ownership of the integrated product outcomes
Lack of a systems vantage point, causes sub-optimized technical solutions (at each site), making downstream integration problematic
Poor, inconsistent communications, causing remote teams to feel isolated and disconnected
Over-reliance on electronic communication tools (tends to drive isolation and lack of confrontation)

Solutions

Core Team; empowered, heavyweight, owns integrated "product" outcome, i.e. from concept to volume production
Core Team rotation; all core team members travel to each location at least once a month
The integrated schedule; score card for each team/location with integration points clearly defined and managed (virtual project)
Refresh Planning; regular and consistent core team meeting time and accountability - use of automation to collect/process progress updates
Include suppliers and key development partners; they are integrated into a single lateral schedule, blurs lines between internal and external resources
Key technical resources visit each site on a scheduled rotation
Use electronic communication tools for asynchronous communications, not as the primary communications mode
Clearly defined self-contained work packages for each team/site with clear interfaces between teams/sites defined

Multiple Companies

Characteristics

Each company has an independent agenda and priorities pulling them in different directions
Team members assigned to project are not empowered to make decisions/commit resources
Team members lack the same motivation and incentives
Team members focused on research outcomes rather than product commercialization
Company A does not commit sufficient resources, forcing Company B to make up the difference
Slow and unclear decision-making; decisions forced through each company's hierarchy which is slow and produces variable results
Lack of a clear understanding of the economics of delay across member companies
Unbalanced or unequal division of responsibility/labor, or unclear success criteria

Solutions

Empowered heavyweight core team with clear roles and responsibilities defined; where all the functions are represented from each company/site
Upfront commitment/contract for each member's participation and contribution
A single overall program owner with authority to make decisions and commit resources
A program budget with accountability metrics defined
Cost-of-delay modeled and used in trade-off decision-making, and to express the value of time
Decision-making system; roles in each decision are defined and a structured process for decision-making is followed
Decision-making cycle time is tracked and reflected in the schedule, so each member can see the impact their decision making has on the program
Success criteria for the program defined that includes the contribution of each team member

Multiple Corporate Stakeholders

Characteristics

Unclear motivations; from very motivated to distracted and disinterested
Lots of "cooks in the kitchen" and unclear who is in charge
A lack of clear product requirements
Conflicting product requirements; customer use-cases, market segments, etc.
Lacking a clear product definition; product must be everything for everyone
Strategic shifts are common during the project; causing confusion and delay
Unclear targets; market windows, success criteria, budget and investment (each stakeholder has their own version)

Solutions

A "Contract" between the stakeholders, like a joint venture project in the construction industry where each stakeholder's role is clear
A single empowered core team with each stakeholder represented; the core team is empowered to make decisions, spend money, chart their own course
A VOC process to determine market and customer requirements; continuously refined through customer council's providing iterative feedback
VOC process owned by the core team
Executive Steering Committee made up of most senior executive from each stakeholder; tie breaker when core team can't decide
Clear target windows set and trends tracked with early warning given through Core Team to Steering Committee

Simultaneous technology innovation and product development

Characteristics

Mindset that innovation can't be managed, it just happens when it happens
Technology and product development are not synchronized, i.e. technology lags behind when it is needed in the product (delay)
Assumption that most of the work requires "innovation"
Resources doing development are also required to innovate
Lack of dedicated resources to the program; not enough subject matter expertise internally, unwilling to seek them outside company
Fear of IP leakage drives inward thinking, lack of solution finding because of people are thinking the same way (group think)
Poor strategic planning of the technology roadmap; forces product development teams to do their own Research while developing
Development projects over burdened with requirement to commercialize while also innovating

Solutions

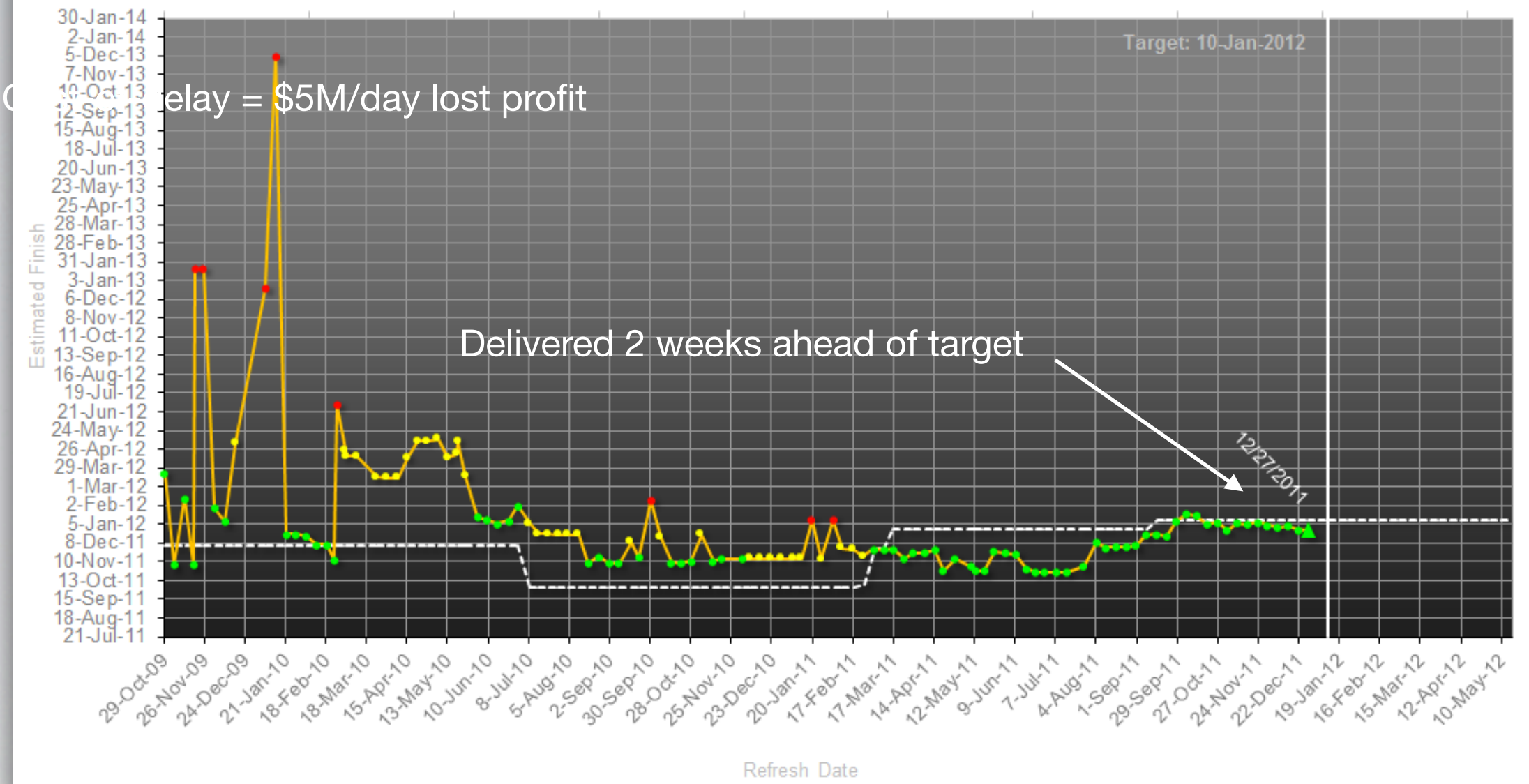
Isolate the "innovation" that is needed; then dedicate colocated team of experts to work on it, insulated from daily interrupts
Define the project such that less than 10% requires innovation/breakthrough thinking (best practice projects have 5-10% new things, rest is known)
Technology roadmap planning, separated from development projects with clear points and times for intersection, i.e. asynchronous streams
Liberal use external subject matter experts, because time to market is more valuable than IP leakage & outside ideas can open minds to new thinking
Cannibalize your own products; the hedge against IP leakage (your competition is always 1-2 steps behind)
Clearly define what is required to commercialize a product quickly and unburden development teams with research requirements
Manage innovation using Learning Milestones (vs product performance milestone); plan learning cycles and use to improve predictability
Innovation = faster cycles-of-learning-increase frequency of learn/fail Development = faster performance maturity-increase rate of product maturity

other slides

\$5B fab and bleeding edge process technology start-up

lateralworks wiggglechart

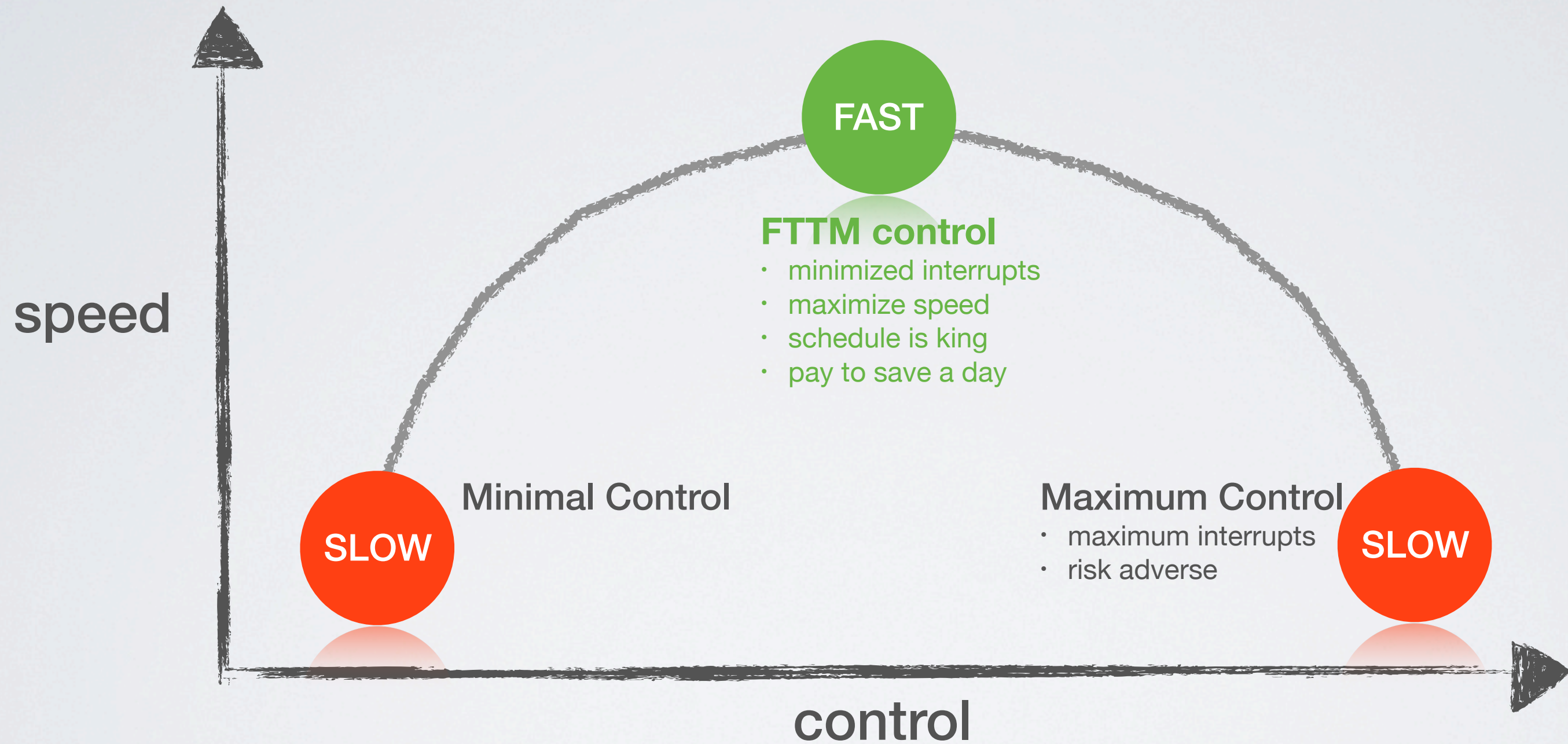
Target Milestone	Target	Estimated Finish	Completed On	Gap (days)	Health
First Silicon Starts	10-Jan-2012		27-Dec-2011	Early: 14d	●



Case study example illustrating how the system works to accelerate complex programs, to close schedule gaps to finish on time

- Complex schedule with over 100k tasks
- Macro-micro schedules with daily Refresh Planning and module level (Etch, Diffusion, etc.)

Control vs Speed



Planning Concepts

- End in mind; no constraints
- Macro to micro, detail over time
- **Realistic planning;** identify gap
- Cause urgency; before-the-fact action taking
- Engage team; they will accelerate
- **Refresh Planning;** continuous pull-in

Refresh Planning; continuous pull-in

