

[Guidelines] Project Planning

Overview	2
One of KIS's key values: Making and Keeping Promises.....	2
What is a plan?	2
Think of your plan like getting to the airport on time	3
There are 2 types of plans	4
Best practices	4
Communicate in terms of dates	4
Work backwards	5
An example of "work backwards" for software:	5
Be ready to change you plan	6
When your plan changes, communicate!	7
A plan doesn't have to be complicated	7
Bare minimum of a plan	7
Simple example.....	8
Plan to get it wrong	8
Give yourself Extra time.....	8
Don't use every day up until the due date	9
Always make a plan, even if you don't have all the details.....	9
Once you have your plan, use your plan	9
TODO: Be careful of blurting out dates wout a plan	10
Design and planning go hand in hand	10
It doesn't matter if you are running (Scrum, Agile, Kanban, Gohorse, ...) you still need a plan	10
Consider the assumptions when making a plan	11
Dependencies Kill projects.....	11
Common Planning Issues	12
Project Planning specific guidelines and templates	13
Estimation	13
Capacity planning.....	13
Dependency tracking.....	13
Project change	13

Task Tracking.....	14
Issue and Risks	14
RACI.....	14

Version Control

Version	Created / Edited by	Summary of changes	Date
V0.1	Ben	Initial Draft	08/15/2023
V0.2	Ben/Marina	Draft review and additional topics	08/25/2023
V0.3	Henrique/Marina	Final Review	09/08/2023

Overview

Project planning is the process of defining the objectives and goals of a project, determining tasks and resources needed to achieve those goals, and creating a timeline for project completion.

Having a project plan in place ensures a smarter way of working. The purpose of a project plan is to provide direction throughout the duration of the project. At the very least it should cover the **what, who, and when** details of the project. It should be something that you, your business and your team **trust on**.

One of KIS's key values: Making and Keeping Promises

- Making date commitments to customers on “when things will be done” is making promises
- “Hitting those dates” is keeping promises.

If we are not talking about dates, we are not making promises

If we are not hitting dates, we are not keeping promises

What is a plan?

- A plan is considering “what needs to be done”
- And “when it needs to be done”
- And maybe “who is going to do it”

It's honestly not more complicated than that

Keep it simple

Think of your plan like getting to the airport on time

Scenario to consider:

- I need to get to the airport by 1 pm
- It takes 40 minutes to drive to the airport
- There might be traffic, so I'll leave at 12 pm.
- It takes about 30 minutes to call the uber, so I'll call the uber at 11:30 am
- I need to eat breakfast, that takes about 1 hour; so I'll start eating no later than 10:30
- I need to pack my bags in the morning, that takes 30 minutes, so start packing at 10:00 am
- I need about 30 mins to shower and dress, so start my shower at 9:30
- I like a little time in the morning to read Instagram, so I should get up no later than 9:00 am

So, your plan would be:

- 9:00 AM – Get up
- 9:30 AM – Take shower
- 10:00 AM – 10:30 - pack bags
- 10:30 AM – 11:30 AM – Breakfast
- 11:30 AM – Call Uber
- 12:00 – 1:00 PM Drive to airport
- 1:00 PM at the airport!

Some lessons from this analogy when planning simple daily life goals or a project

- You need to think through all the “tasks”
 - Eat breakfast, pack bags, take shower, ...
- You need to understand sequencing
 - I can't take my shower after I pack my bags
- You need to understand how long each task takes:
 - 30 min shower, 40 mins drive to airport
- You need to consider risk, and adjust
 - There might be traffic, so use 1 hour instead of 40 mins to drive to airport
- Understand the “due date” (1 PM) and WORK BACKWARDS

There are 2 types of plans

High-Level Plan:

- Strategic focus
- Long-term perspective (months to years)
- Minimal detail on execution
- Flexibility and adaptability
- Communication and alignment of vision
- Usually created before designing a feature
- Dates may change

Detailed Plan (Delivery Plan):

- Tactical focus
- Short to medium-term perspective (weeks to months)
- Comprehensive detail on execution
- Rigidity once set in motion
- Task-level accountability
- Usually created/reviewed after designing a feature
- Committed dates

Best practices

Communicate in terms of dates

We always need to be talking about dates.

When discussing development tasks, deliverables, features, or process steps, it's crucial to always include specific dates in the conversation. This applies to your status reports (for more details on status report guidelines, check [here](#)), internal communications, as well as interactions with clients.

Few examples of good and bad communications:

- Bad: "We finished XYZ and we will be working on ABC next"
- Good: "We finished XYZ and will be starting work on ABC on 12/15"
- Better: "We finished XYA and will be starting work on ABC on 12/15, our target complete date is 1/20"
- Best: "We finished XYZ and will be starting work on ABC on 12/15, our target complete date is 1/20. QA should start Jan 10. We will need requirements signed off by 12/20"

Work backwards

- In a perfect world, you have all the tasks, and a good estimate for each task and then you would “make the due date”
- However, in the real-world, very often you “have the date already”
 - *I need to get to the airport by 1PM*
- There is some business reality or promise that drove the date
- Or maybe there was a high-level plan already communicated with a date
- So, you need to “work backwards”
- Start with your “goal” -> “Deliver the Software”
 - And think through each step backwards.
- You need to determine your “required start date” for each task to “hit the end date”
 - [See Airport Analogy](#)
- Often when you do this, you might determine “oh my gosh I am already late!?”
 - I had this exact conversation with someone trying to get to the airport on time once...
 - We worked backwards and he realized: “I am already late!”

An example of “work backwards” for software:

Consider:

- My target go-live date is 6/15
- I should plan for a production pilot, with a few users and some time to make fixes in prod:
 - So, I should be in prod on 6/1
- QA should take about 2 weeks
 - So, I should start QA on 5/15
- But wait, if QA finds something, I'll need time to fix it also, maybe 2 weeks.
 - So, I should start QA on 5/1 and plan to fix bugs from 5/15 – 6/1
- It will take me about 6 weeks to write the code
 - So, I'll start code writing on 3/15
- I'll need about 1 week of design
 - So, start the design on 3/7
- I'll need about 2 weeks of analysis and requirements documentation
 - So, I should start 2/20

Your plan would be:

Start	End	Task
20-Feb	3-Mar	Analysis and documentation
7-Mar	14-Mar	Design
15-Mar	30-Apr	Coding
1-May	15-May	QA
15-May	1-Jun	Find fix
1-Jun		Prod soft launch
1-Jun	15-Jun	Production pilot
15-Jun		Full Launch

Be ready to change you plan

- Plan are always changing, so be rady for it
- In the airport analogy, what should you do if you are already running late?
- You may use the “The three-legged stool”
 - You can fix one leg and move 2
- Every project is like a “The three-legged stool”
 - conceptual framework that represents the three fundamental and interconnected aspects of managing a project effectively.
 - Scope, time and resources
- When you’re running late, you need to change one or more legs
- For instance, if there's a resource shortage, temporarily reducing project scope can ensure timely delivery while maintaining equilibrium.
- Clear communication with stakeholders is essential when removing a leg to explain the rationale and the impact on project objectives.
- Sometimes, due to unforeseen circumstances or strategic shifts, the manager may need to temporarily or permanently remove one leg of the stool.

Adding People:

- To accelerate a project or handle a sudden workload increase, adding team members can extend the 'resource management' leg.
- Considerations include effective coordination, potential temporary cost increase, and adjustments to the project plan.
- Communication with the team and stakeholders is critical when adding people to the project.

Extending Time:

- Extending the project timeline can accommodate scope changes and provides flexibility in managing 'time management.'

- Such extensions should be communicated to stakeholders to manage expectations and ensure understanding of the rationale.
- Recognize that extending time may affect project costs and resource availability.

Scope management:

- Defines and controls the project's scope
- Define what could change in the scope to adjust to the availability of your plan
 - For example, take something off, break the scope into minor deliverables
- Encompasses all project work, objectives, deliverables, and features.
- Aims for a clear understanding of project boundaries.
- Prevents scope creep, uncontrolled expansion of project scope.
- Maintains focus on original project goals.

Maintaining Equilibrium:

- Adjusting one leg of the stool can impact the others, creating a ripple effect.
- Project managers must carefully assess the implications of adjustments and make informed decisions.
- Effective communication is vital to keep stakeholders informed and aligned with any changes to the three-legged stool, ensuring project success.

When your plan changes, communicate!

Plans are never perfect

- Dates frequently move
- Of course, we want to try to minimize this, but it happens.
- When it does: COMMUNICATE
- *Don't surprise people!*

Look at the Status Report guidelines [here](#)

Look at project change guidelines [here](#)

“People don't like bad news, but they don't like surprises even less.”

A plan doesn't have to be complicated

Bare minimum of a plan

- “The understood scope”
 - A list of deliverables, or ideally technical tasks.
 - You need the list of “things to do”
- “An estimate on each task”

- Each task needs an estimate
 - Even a high-level estimate is better than no estimate
- “A target date for each task”
 - When should each task be done to achieve the larger goal.

Simple example

- This is a plan:
 - 12/1 Start analysis / requirements
 - 12/30 requirement and design complete
 - 1/5 development start
 - 2/15 QA Start
 - 3/1 QA find / fix
 - 3/15 UAT Start
 - 3/30 UAT find / fix start
 - 4/15 Production Deploy
 - 4/20 Production Pilot Start
 - 5/6 Pilot find / fix
 - 5/15 Full Rollout
 - 5/15 – 6/15 Hypercare support
- The above list of tasks and dates could quite literally be used to keep a simple project on track.
- Using simple milestone dates provides clarity to the business on “what to expect when”
- Provides a set of targets for the dev team to hit
- Even a simple plan like the one above could address many of our current issues

Plan to get it wrong

- Your plan won't be perfect
- So, give yourself extra time
- Don't use every day up until the due date
- And add specific “I got it wrong” tasks.
- For example: be sure you plan for dev time to “fix bugs” after QA
- Plan for a “soft launch” to see in production before the due date...
- Don't assume everything will be perfect on your finale due date or in plan dates...

Give yourself Extra time

- Put pad in your plan

- No matter how good you think your estimates will be
- You will always need extra time
- Maybe you forgot a task?
- Maybe you got “stuck” on a task?
- You will always need more time
- Especially give yourself extra time on the RISKY parts of the plan

Don't use every day up until the due date

- If your target date is 12/15, then plan to be done on 12/1
- This is a specific example of “Give yourself extra time”
- Why do this?
 - Because you should “plan to get it wrong”

Always make a plan, even if you don't have all the details

- Even if you don't have all the details, you still must make a plan
- Business people want to know “When can I expect the Flux Capacitor feature to be done?”
- They will want to know this sometimes even before the project starts, even before the budget is approved!
- This is reality! Businesses need to plan ahead
- They need to have date projections sometimes long before you have all the details.
 - This is ok!
- Make your best guess:
 - Put a high-level plan on paper
 - Clearly communicate this is a “high level plan”
 - Meaning: these dates might change until we have more information and make a design
- The **worst** thing you can do is not provide any plan (**don't do this!**)

Once you have your plan, use your plan

- Use your plan to hold you and your team accountable
- Use your plan to hold others accountable too:
 - Your business partners
 - Third parties that you need to help you
- Watch the dates on your plan
- Adjust if needed!
 - See “Be ready to change your plan section”
- Use the plan to “drive a sense of urgency”

- [Try to keep your dates](#)

Be careful of blurting out dates wout a plan

- Mentioning dates in conversation or communication without a clear plan or context can lead to confusion, misunderstandings and false expectations
- Ensure that the dates you mention are accurate.
- Software design is a key step in good planning.
- if you need to give an answer right away, make it clear that it's a high-level estimate, not a promise. And only commit to the delivery date after design, after more thinking and after having all information you need

Comentado [MS1]: @Ben Quintana and @Henrique Fuschini can you check if this guidelines is what you meant ?

Design and planning go hand in hand

- Software design is a key step in good planning.
- The output of the design really should be: THE TASK LIST
- Once you have a good design, and only once you have a good design can you say:
 - All the things you need to do
- And only after you have a good design can you make good guesses about:
 - How long each task will take
- You can and should make **high level plans** before you have designs.
 - For example, you should plan when your design will be complete!
- But you cannot have a “solid plan” or a [delivery plan](#) until you have a design
 - Look into PS design guidelines [here](#)

It doesn't matter if you are running (Scrum, Agile, Kanban, Gohorse, ...) you still need a plan

- No matter what development mythology you are using, you still need dates on a calendar
- “The business doesn't care about your sprints”
- The business just cares about: “When will XYZ highbe done?”
- This is a fair question on their part, this is the real world
- No matter what methodology you are using, you need to think and communicate target delivery dates
- Sprints are good, scrum is good, but your sprints need to tie to larger target dates / milestone dates
- Remember, one of [KIS' core values](#) is to keep “date promises”

Consider the assumptions when making a plan

- Assumptions are statements considered true for planning but not guaranteed
- Common Assumptions: Resource availability, stakeholder support, scope stability, cost estimates, out of scope items, etc.
- Assumptions can become outdated or invalid as the project progresses or external factors change.
- Assumptions are crucial because they help in making decisions when complete information is not available.
- Regularly review and validate assumptions as the project advances, updating the plan as needed.
- Assumptions are closely related to project risks. Unvalidated or inaccurate assumptions can lead to unforeseen risks.

Dependencies Kill projects

Most initiatives require the **collaboration** and **coordination** of **multiple teams**. At the very least, 2 parties are involved in every project: the business, and the IT partner. More complicated deliveries can involve multiple technical teams, external partners, and different business groups within the clients. One of the most frequent problem areas for project delivery is when one team relies upon another team or task for successful completion.

- In the [airport analogy](#) a dependency would be ,for example, have money to pay for the cab or have your passport in hand to travel, etc.
- In a new integration with a third party a dependency would be, for example, to receive a historical load file from the source system before processing it.

What the manager needs to do:

- Regularly review and validate dependencies as the project advances, updating the plan as needed.
- Keep dependency owners accountable and constantly communicate (use [status reports](#) and other channels)
- dependencies are closely related to project risks. Delayed dependencies can lead to unforeseen project changes.
 - See more details about dependency tracking [here](#)

Have effective communication about identified risks with stakeholders

- Risks are events or conditions that have the potential to impact the project's objectives negatively, leading to deviations from the planned outcomes.

- Types of Risks:
 - Internal Risks: These originate from within the project and are often under the project team's control.
 - Examples include scope changes, resource constraints, and technical challenges.
 - External Risks: These stem from external factors beyond the project team's control, such as market fluctuations, regulatory changes, or supplier issues.
- Few Common Project Planning Risks:
 - Scope Creep: Uncontrolled changes to project scope can lead to increased costs and delays.
 - Resource Constraints: Insufficient or misallocated resources can hinder progress.
 - Unclear Objectives: Ambiguity in project goals can result in misalignment and confusion.
 - Budget Overruns: Inaccurate cost estimates can strain financial resources.
 - Scheduling Conflicts: Poorly managed timelines can lead to missed deadlines.
 - Stakeholder Conflicts: Disagreements among stakeholders can disrupt project progress.
 - Technical Challenges: Complex technologies or unfamiliar processes may pose unexpected hurdles.
- Risks evolve over the project's life cycle; continuous monitoring is essential.
- Effective risk management involves identifying, assessing, and mitigating risks associated with key assumptions.
- Risks management in Project Planning is a critical component of success

Common Planning Issues

- The most basic issues:
 - Not making a plan / Not having target dates
 - Not considering all the tasks that need to be done
 - forgetting tasks
 - Not making estimates / bad estimates for tasks
- More complicated issues:
 - Underestimating the work for the tasks
 - Not thinking through the team's capacity
 - Jerry only has 50% of his time for the project
 - Thinking you can use all the time up until the due date
 - Not considering dependencies
 - "A need to be done in order for B to be done..."
 - Not considering risks
 - "What make this plan go wrong"
- As of this writing, KIS is honestly struggling with "the basic issues"

- Our most common issues are not making plans / not making estimates

Project Planning specific guidelines and templates

Estimation

Access the in-depth guidelines by clicking [here](#).

The estimation process tries to achieve the most accurate estimate of what it will take to delivery a specific project, feature or other set of work. To make an accurate estimate the following information is required:

- Business request overview (business scope)
- Some understanding of the technical delivery (what will be developed)
- Assumptions and Dependencies

Capacity planning

Access the in-depth guidelines by clicking [here](#).

Capacity Planning is the process of looking at the allocation of all members of a team across different work activities.

The purpose of Capacity Planning is to ensure:

- Resources are not under or over allocated
- Serves to get a high-level view of “where” resources are working
- Gives a view of where resources should be spending time across concurrent / planned workstreams
- Allows for a forecast of when resources will become available
- Allows for a forecast of when resources may need to be added to a team

Dependency tracking

Access the in-depth guidelines by clicking [here](#).

Most initiatives require the **collaboration** and **coordination** of **multiple teams**. At the very least, 2 parties are involved in every project: the business, and the IT partner. More complicated deliveries can involve multiple technical teams, external partners, and different business groups within the clients. One of the most frequent problem areas for project delivery is when one team relies upon another team or task for successful completion.

Dependencies kill projects and keep promises

Project change

Access the in-depth guidelines by clicking [here](#).

Changes are natural in any project. Even though when we think about a project change, the first things that come to mind is a scope change, however that's not all that may change during the lifetime of a project. A few examples can be seen below:

- Changes to your target delivery date and project milestones.
- Technical difficulties that create a delay in the implementation.
- One of the assumptions that were originally pointed out are no longer valid
- Change in priorities.
- People leaving the project and/or new people joining the team.

Task Tracking

Having a way to track your team's tasks and progress is essential to make sure your project is moving well and expected dates will be hit; it also highlights individuals' performance without having to micro-manage anyone; it helps you taking decisions about delegation, prioritization and anticipates relevant information for you and the client

- Keep track of what your team is doing
- Have a task tracking tool
- You still need a project plan

Access the in-depth guidelines by clicking [here](#).

Issue and Risks

Issues and risks are critical concepts in project planning and management. They are essential because they help project managers anticipate, identify, assess, and address potential challenges that can impact a project's success.

They are potential events or situations that could negatively impact the project in the future. They are uncertainties that, if realized, can affect project scope, schedule, cost, or quality.

- Identifying issues and risks during planning enables proactive problem-solving.
- Issues and risks can lead to cost overruns and schedule delays if not addressed in planning.
- Transparent communication about issues and risks helps manage stakeholder expectations.
- Effective management of issues and risks increases the likelihood of project success.

Access the in-depth guidelines by clicking [here](#).

RACI

RACI is an acronym that stands for Responsible, Accountable, Consulted, and Informed.

It is a project management tool used to define and clarify roles and responsibilities within a project or an organization. RACI charts, also known as RACI matrices or RACI diagrams, are created to specify who is Responsible for tasks, who is Accountable for the overall outcome, who needs to be Consulted for input or expertise, and who needs to be Informed about progress or decisions.

- RACI charts provide clarity by clearly defining who does what in a project. This reduces confusion, overlaps, and gaps in responsibilities.
- Designating someone as Accountable ensures that there is a single person responsible for the success of each task or the project as a whole. This enhances accountability and decision-making.
- By specifying roles, project managers can allocate resources more efficiently, ensuring that the right people are assigned to the right tasks.
- RACI charts facilitate better communication by outlining who needs to be Consulted or Informed. This streamlines communication channels and ensures that key stakeholders are kept in the loop.

Access the in-depth guidelines by [clicking here](#).