

# User Story Mapping

and three strategies for managing uncertainty

Nils Christian Haugen  
Wasteless AS

wasteless



a common problem,

a common problem,  
and three strategies to deal with  
it



The stories you are about to hear  
are made up.

The stories you are about to hear  
are made up.

Only the names have been left  
unchanged to offend the innocent.





This is **Roger**





This is **Roger**

He's a seasoned business person.



## This is **Roger**

He's a seasoned business person.

(he's looking pretty seasoned)



## This is **Roger**

He's a seasoned business person.

(he's looking pretty seasoned)

*“This XP stuff makes a lot of sense!”*



## This is **Roger**

He's a seasoned business person.

(he's looking pretty seasoned)

*“This XP stuff makes a lot of sense!*

*I just break my requirements down into user stories, and the team builds them one at a time.*



## This is **Roger**

He's a seasoned business person.

(he's looking pretty seasoned)

*“This XP stuff makes a lot of sense!*

*I just break my requirements down into user stories, and the team builds them one at a time.*

***It's all just like adding bricks to a wall!”***



This is **Melanie**







This is **Melanie**

She works on the development team



This is **Melanie**

She works on the development team

(she's good with people)



## This is **Melanie**

She works on the development team

(she's good with people)

*“Roger, let's get your requirements down as stories.”*



## This is **Melanie**

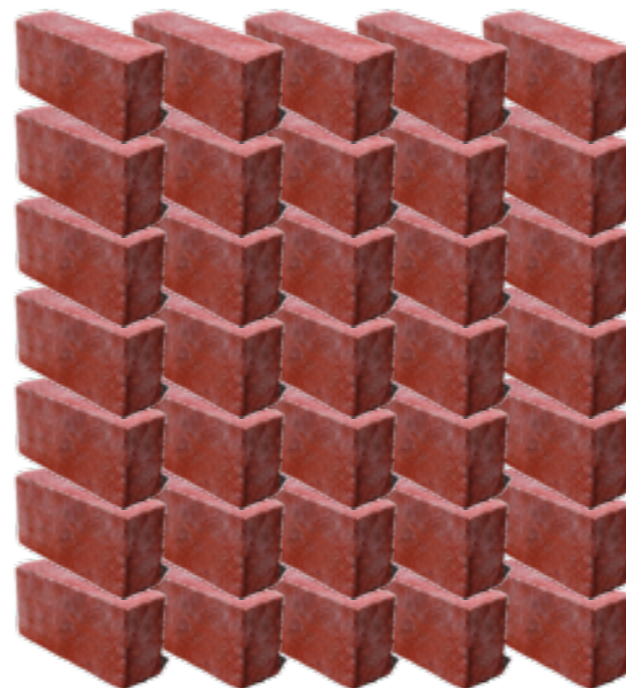
She works on the development team  
(she's good with people)

*“Roger, let's get your requirements down as stories.*

*You're the expert. So, tell me what you want.”*

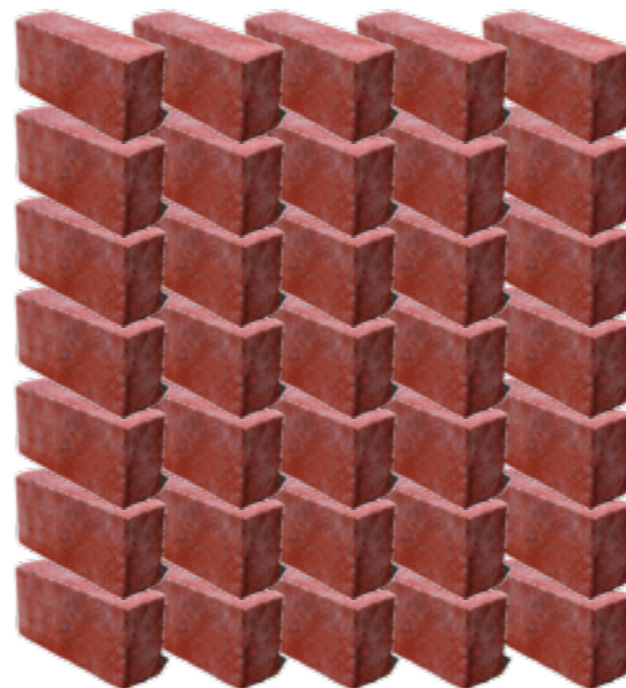


Roger & Melanie work together to capture stories. Then after estimating Roger chooses the highest value stories for the first release.



Roger & Melanie work together to capture stories. Then after estimating Roger chooses the highest value stories for the first release.

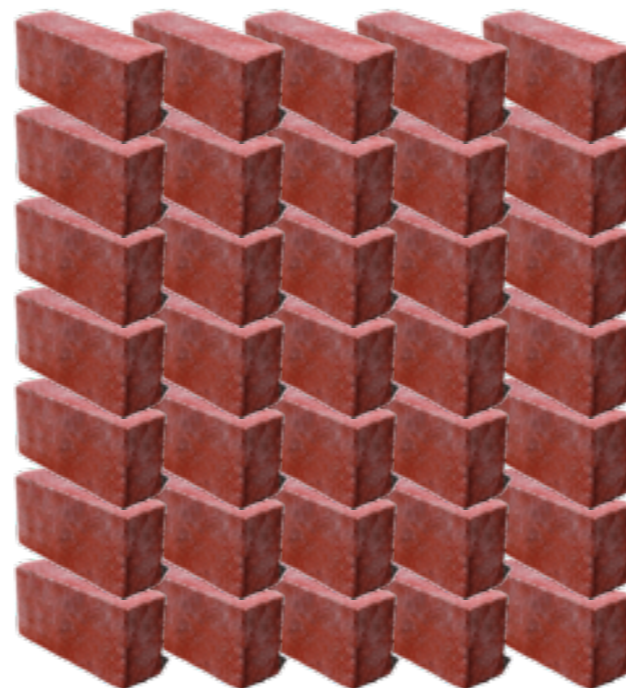
*“Melanie, I really need **all** this stuff on time or my stakeholders will kill me.”*



Roger & Melanie work together to capture stories. Then after estimating Roger chooses the highest value stories for the first release.

*“Melanie, I really need **all** this stuff on time or my stakeholders will kill me.”*

*“No worries Roger.”*



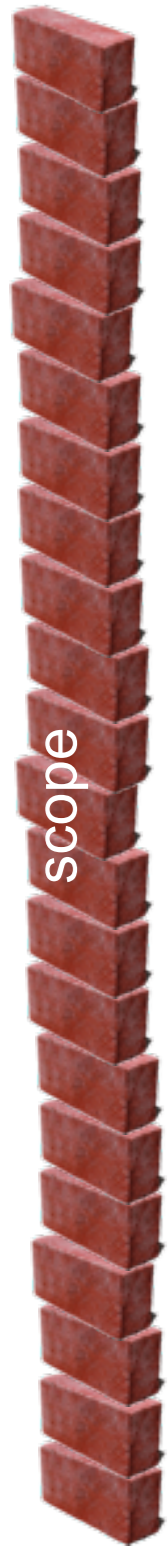




Roger learns to use a burn-down  
chart to monitor the progress of  
Melanie and her team

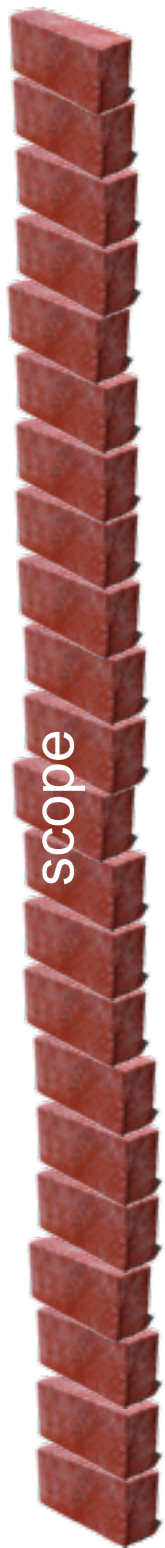


*“At the end of each iteration, I just count how many bricks, uh... stories are left. It couldn't be more simple!”*



scope





scope



1

2

3

4

5

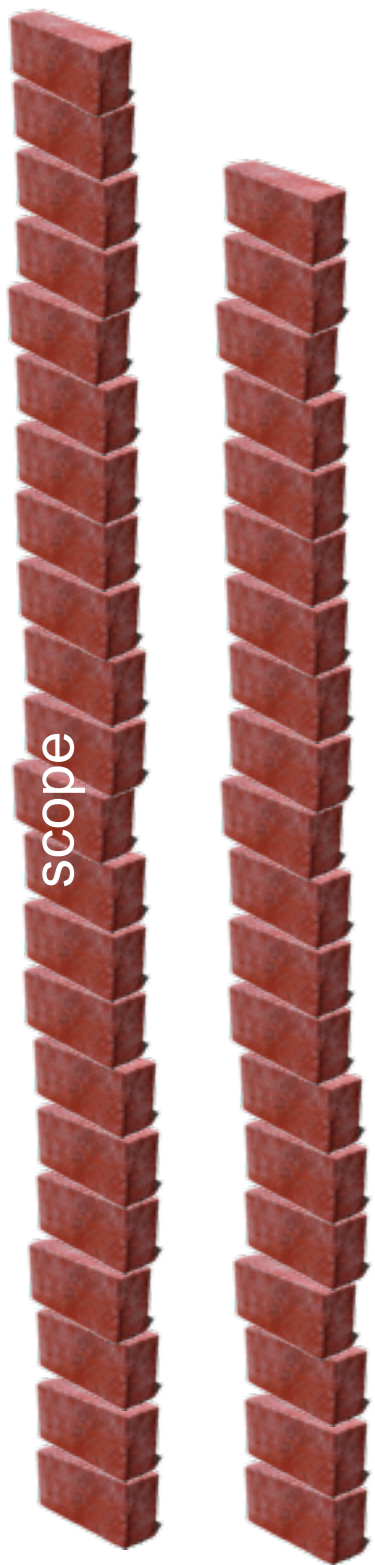
6

7

8

*“Iteration 1.... things are going fine.”*





scope



iterations

1

2

3

4

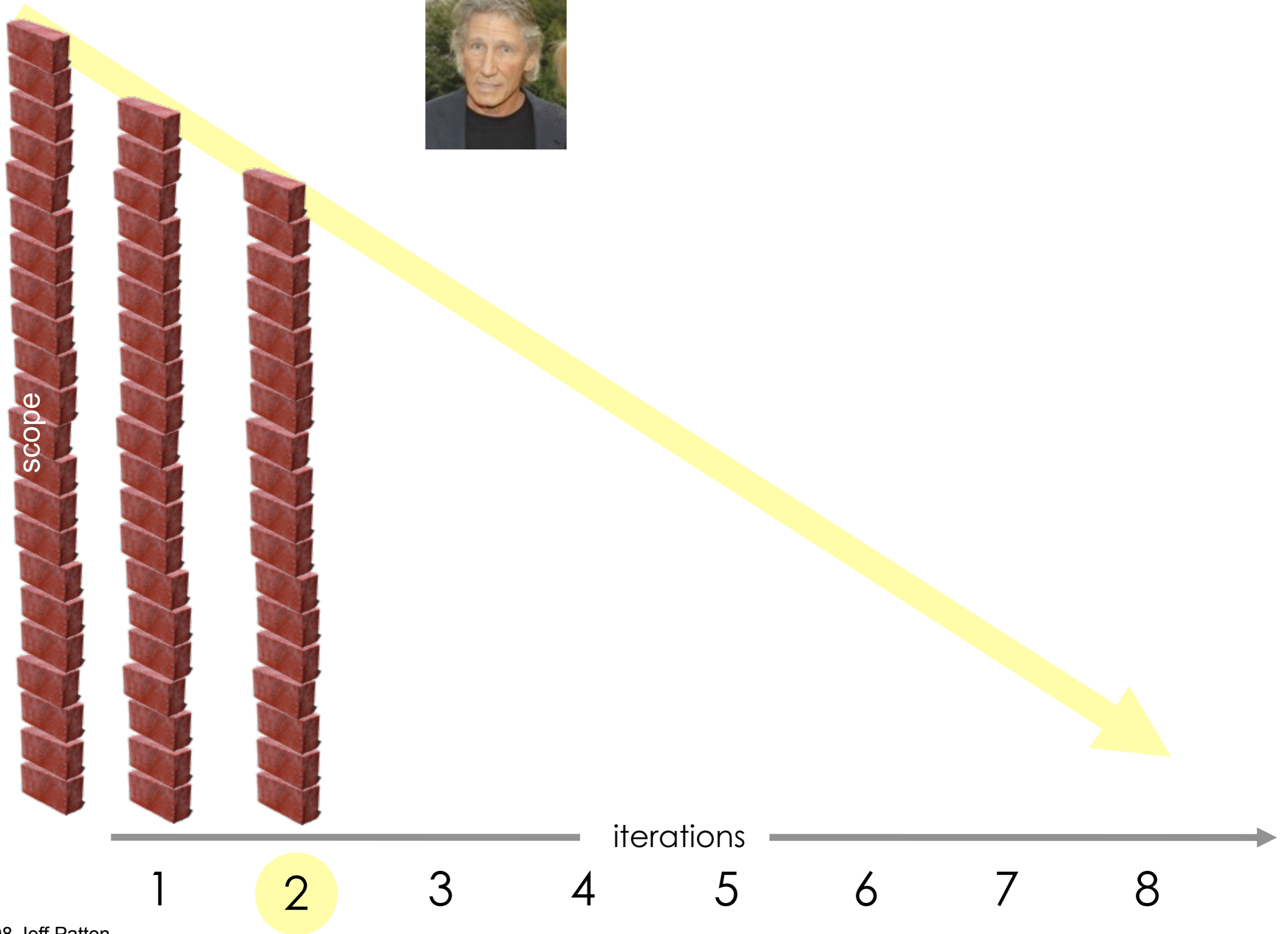
5

6

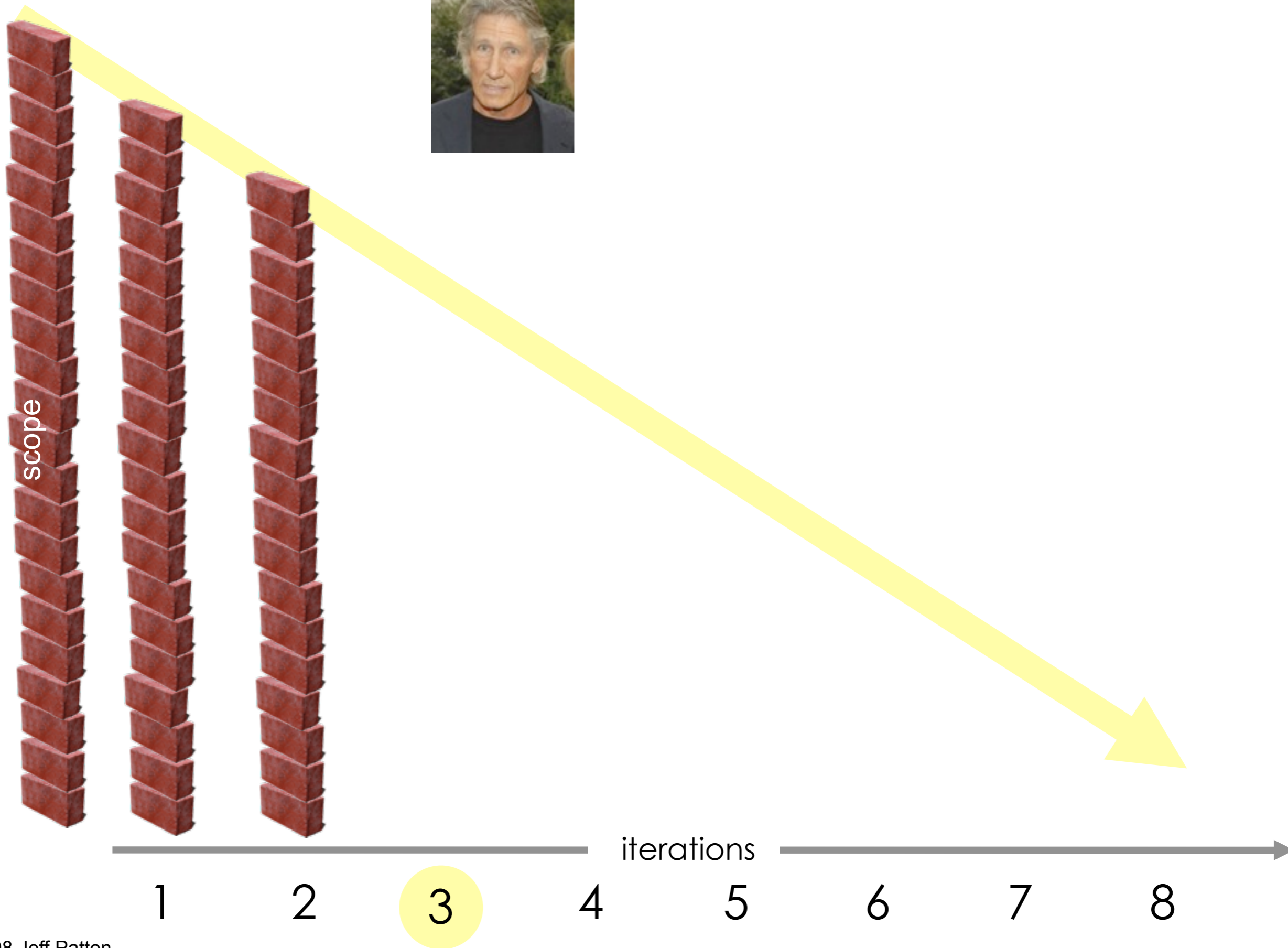
7

8

*“Iteration 2.... the trend looks fine.”*

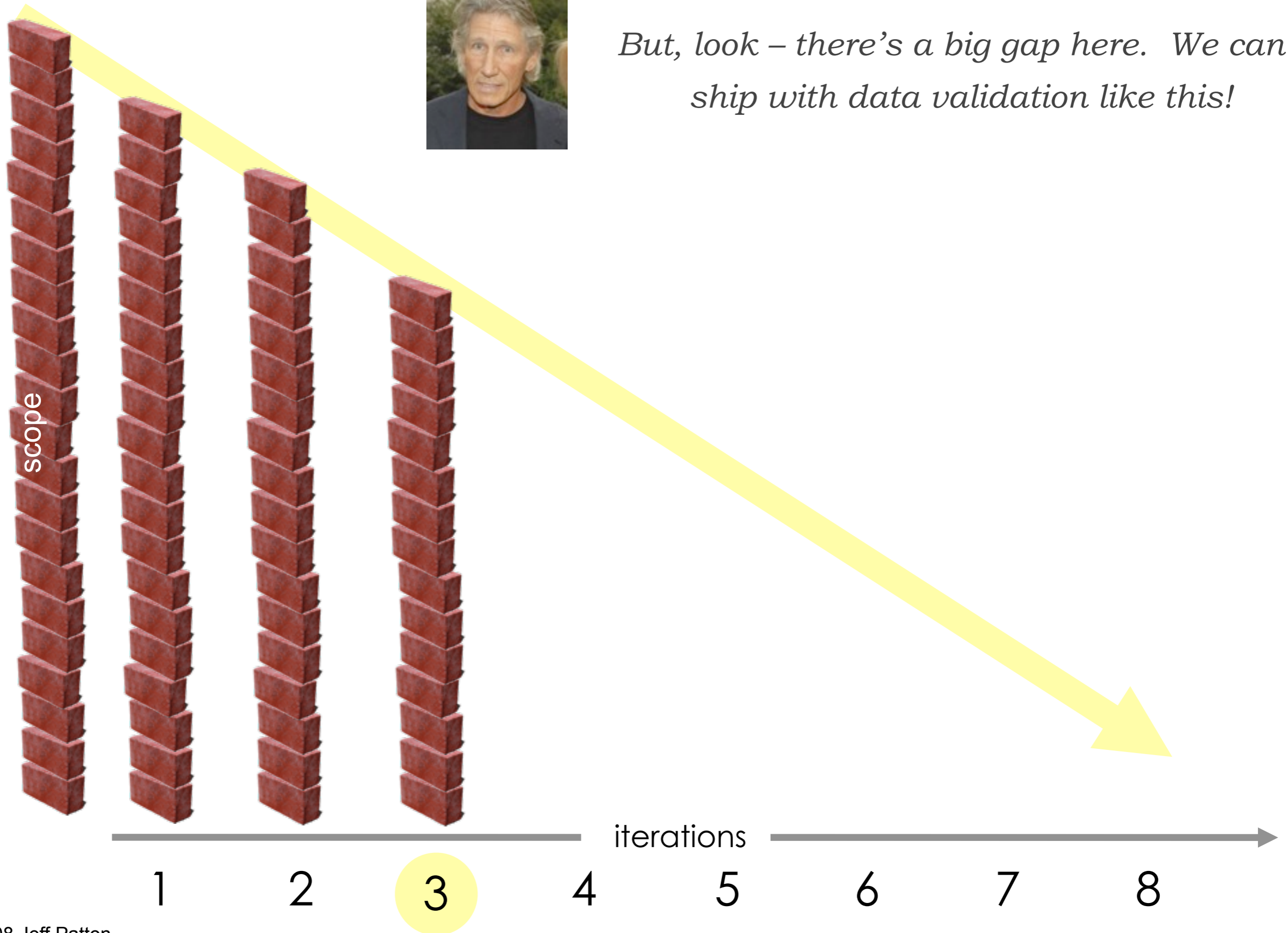






*“Iteration 3 – velocity is getting even better!”*

*But, look – there’s a big gap here. We can’t ship with data validation like this!*

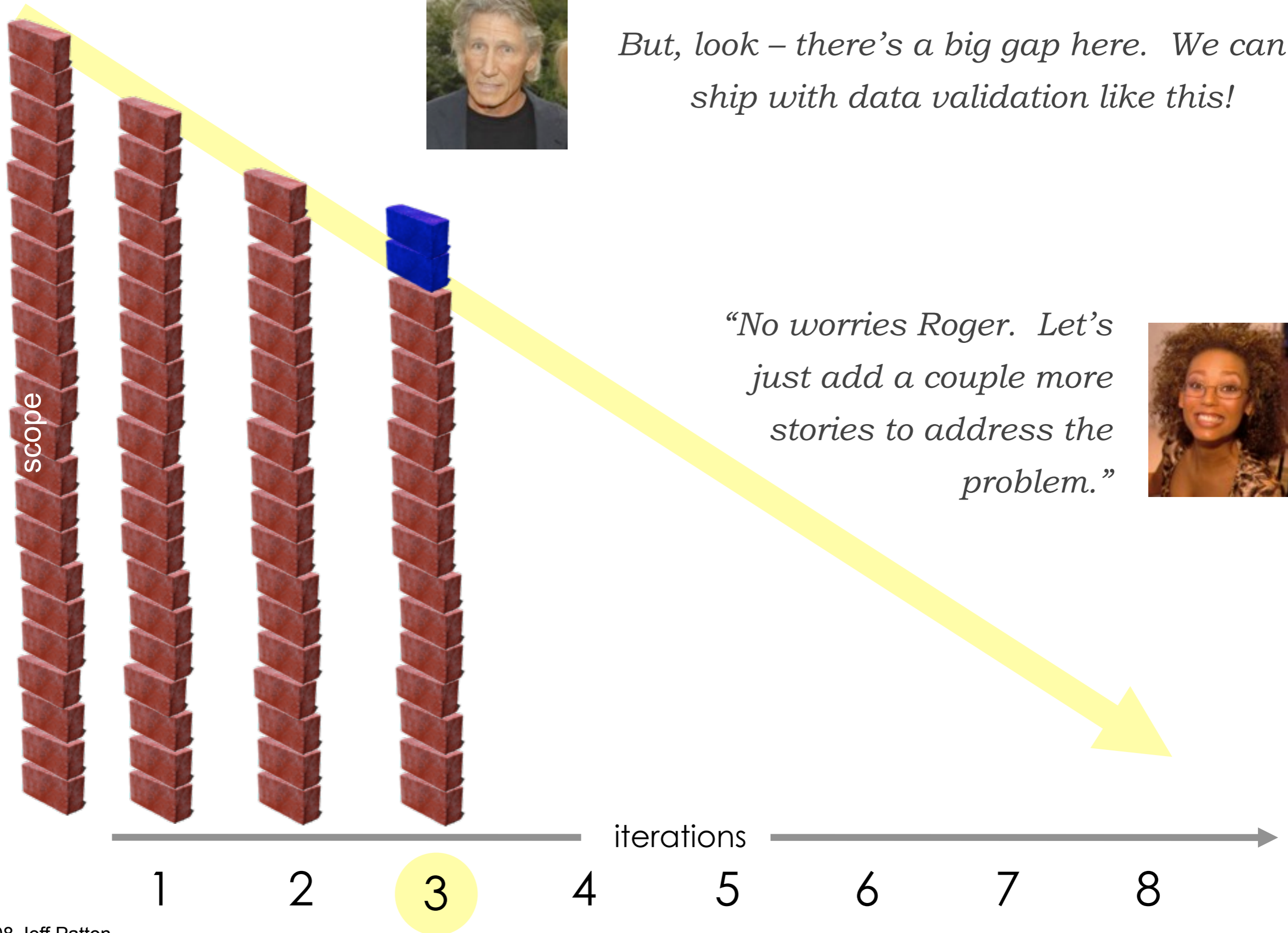


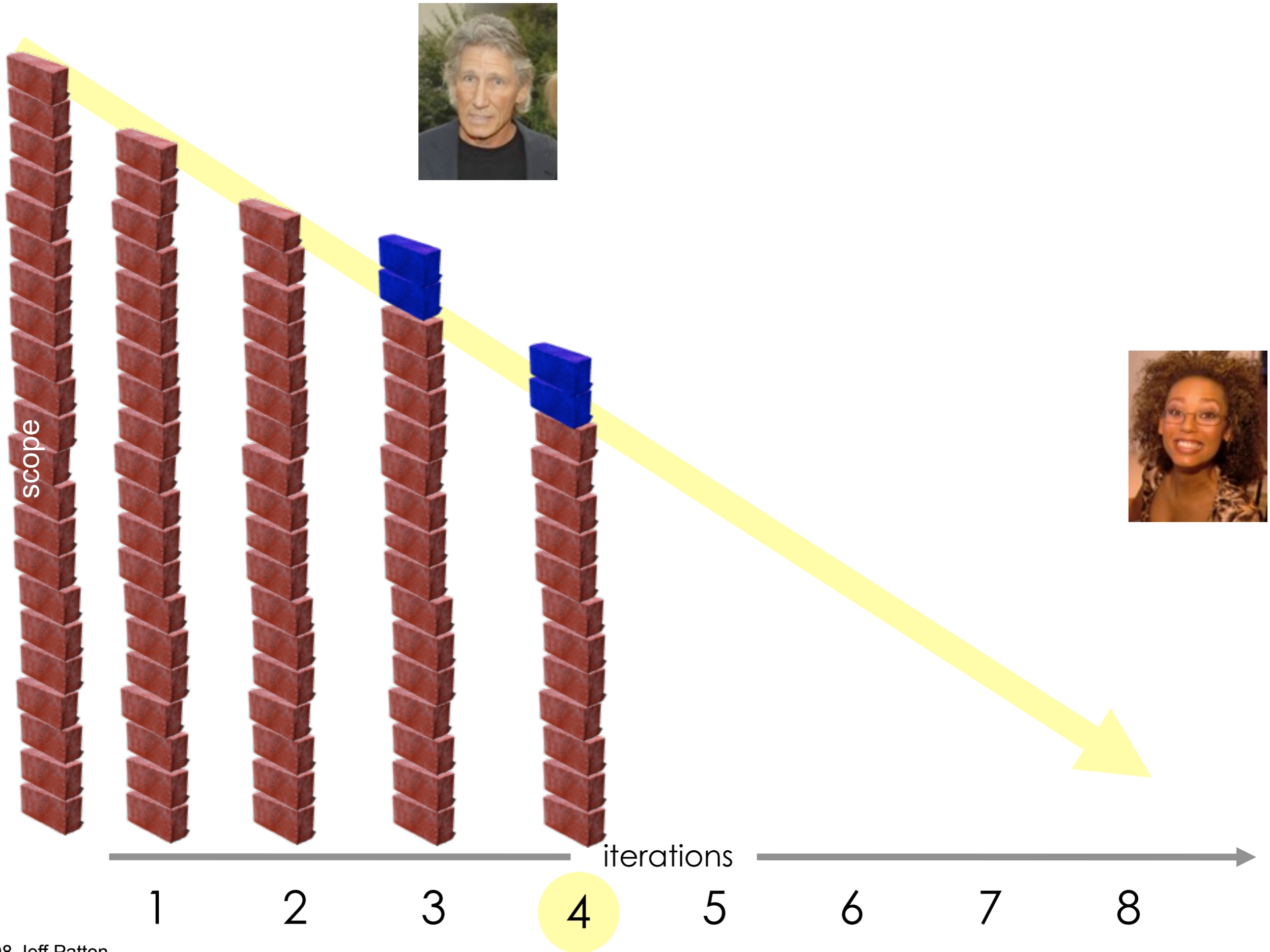
*“Iteration 3 – velocity is getting even better!”*

*But, look – there’s a big gap here. We can’t ship with data validation like this!*

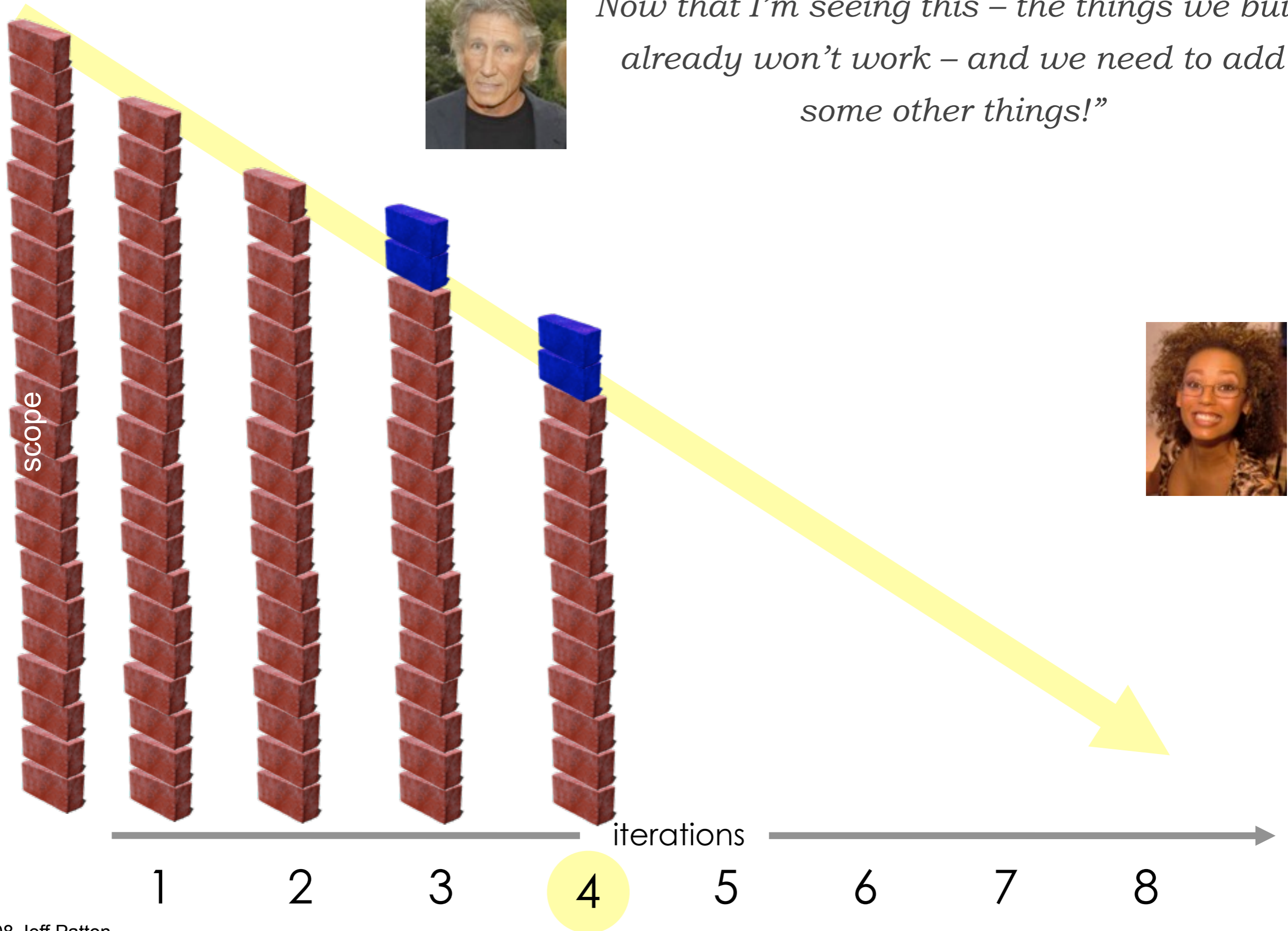


*“No worries Roger. Let’s just add a couple more stories to address the problem.”*





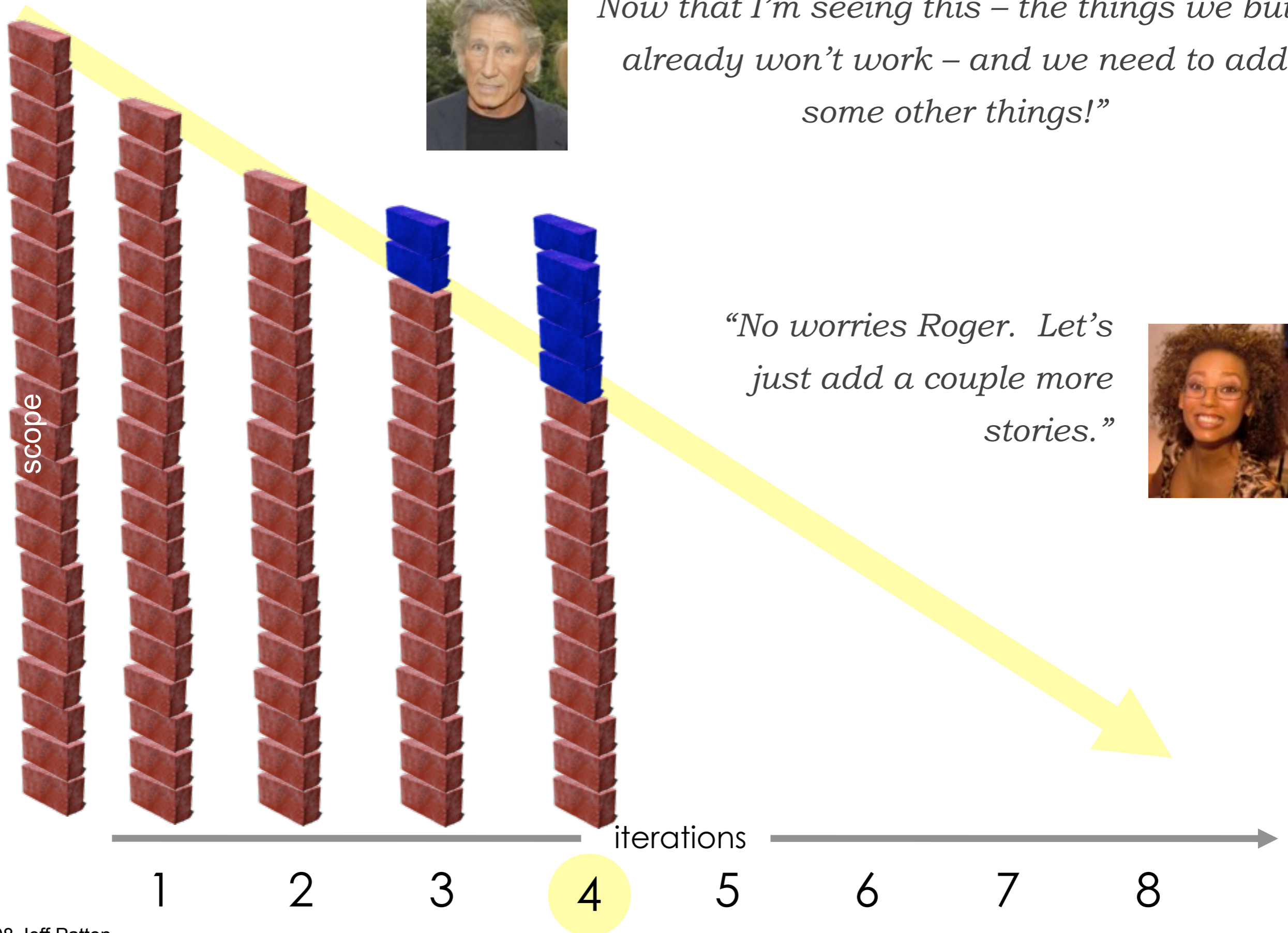
*“Iteration 4. We’ve got real problems here. Now that I’m seeing this – the things we built already won’t work – and we need to add some other things!”*

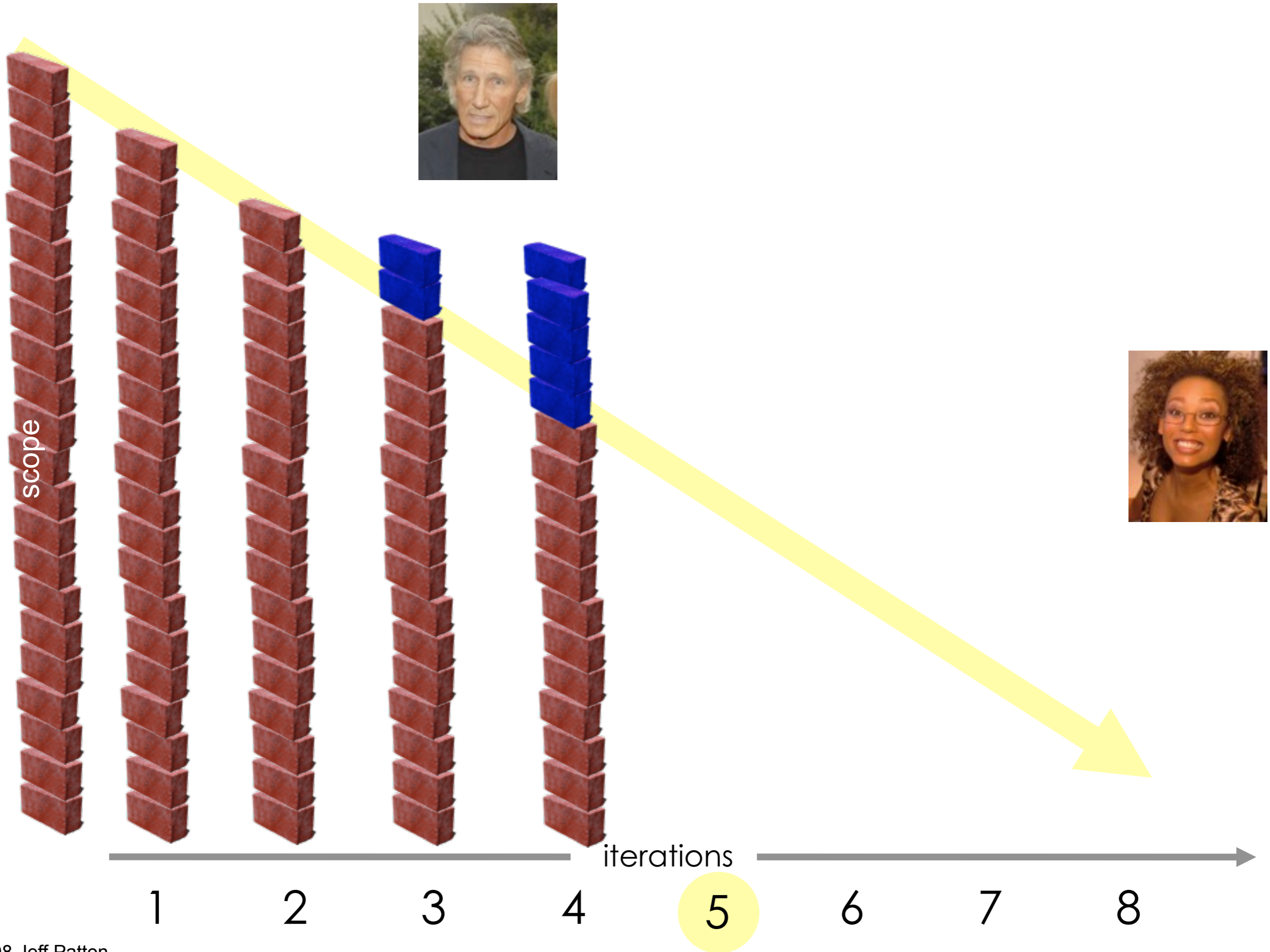


*“Iteration 4. We’ve got real problems here. Now that I’m seeing this – the things we built already won’t work – and we need to add some other things!”*

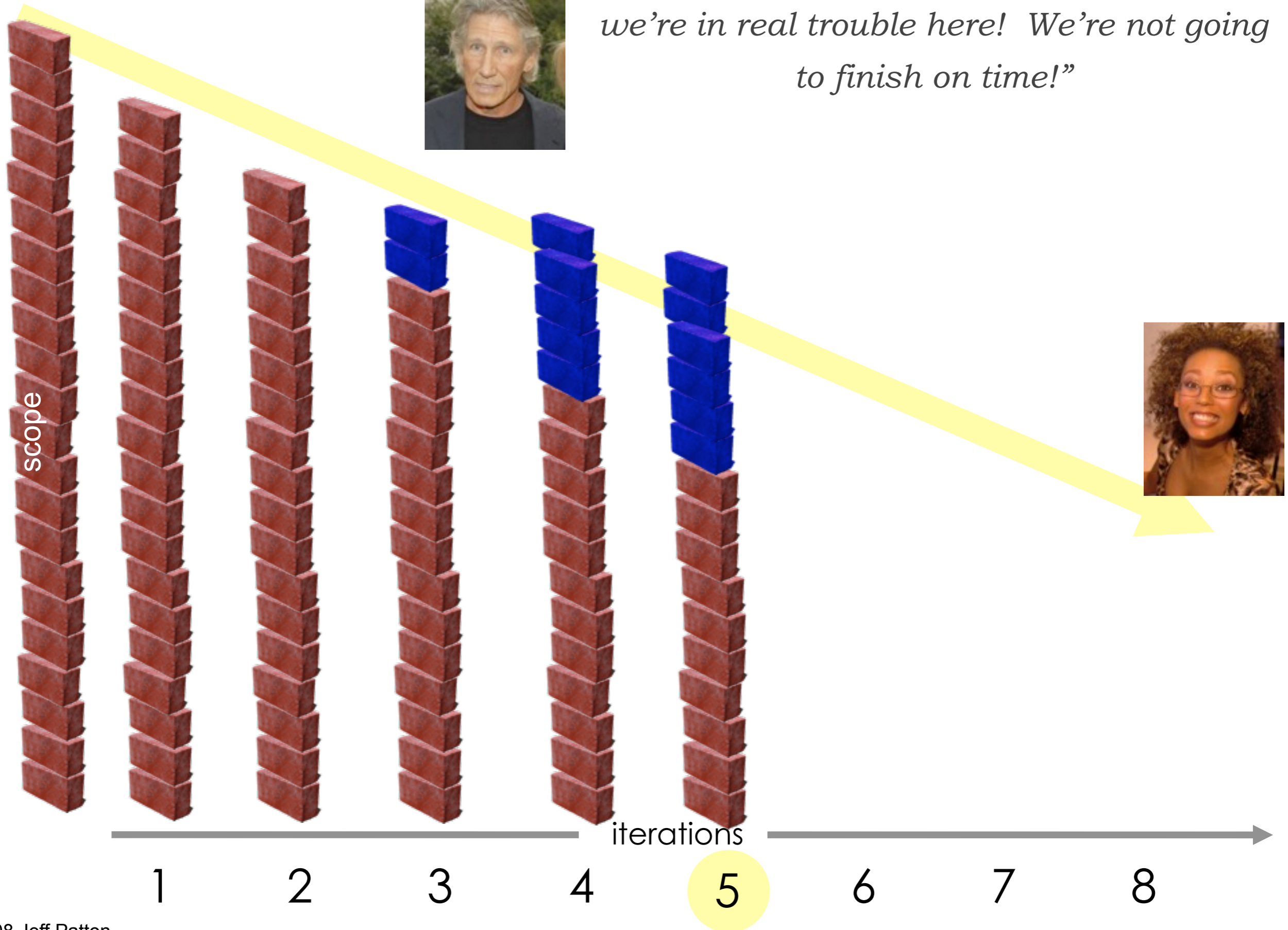


*“No worries Roger. Let’s just add a couple more stories.”*



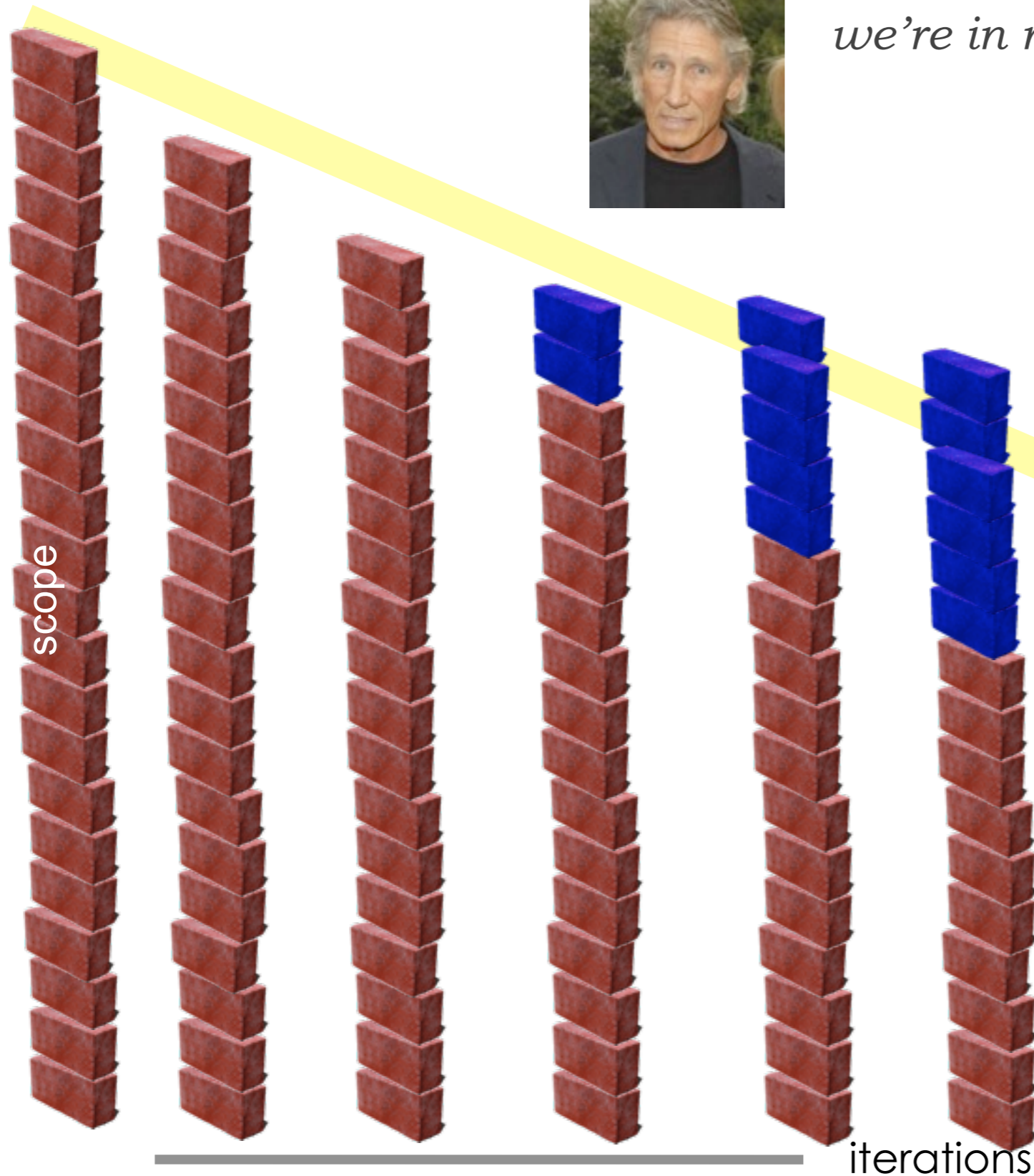


*“There are still more holes... Look – we’re in real trouble here! We’re not going to finish on time!”*





*“There are still more holes... Look – we’re in real trouble here! We’re not going to finish on time!”*



*“No worries Roger. Let’s just drop a few stories.”*



1

2

3

4

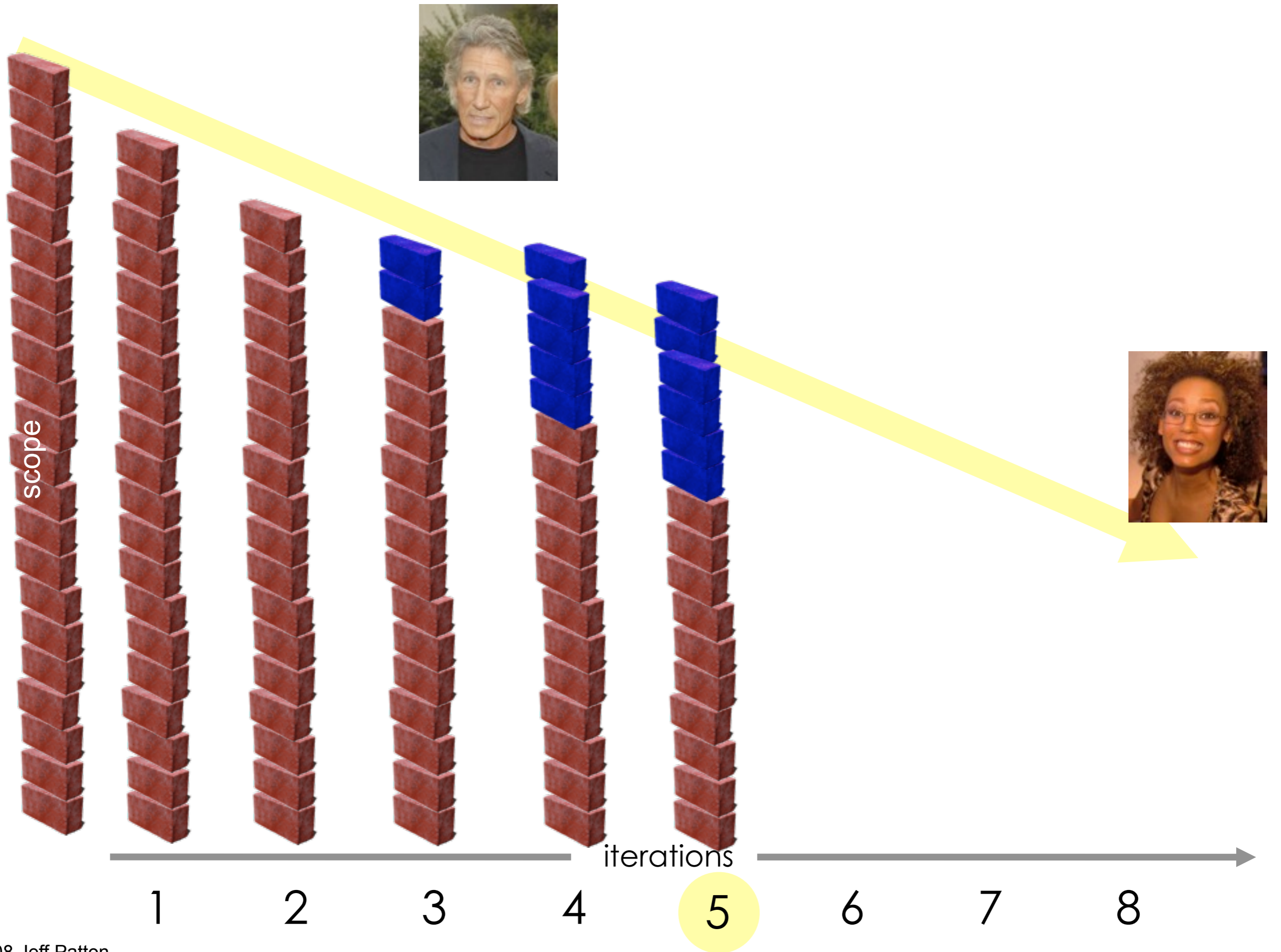
5

6

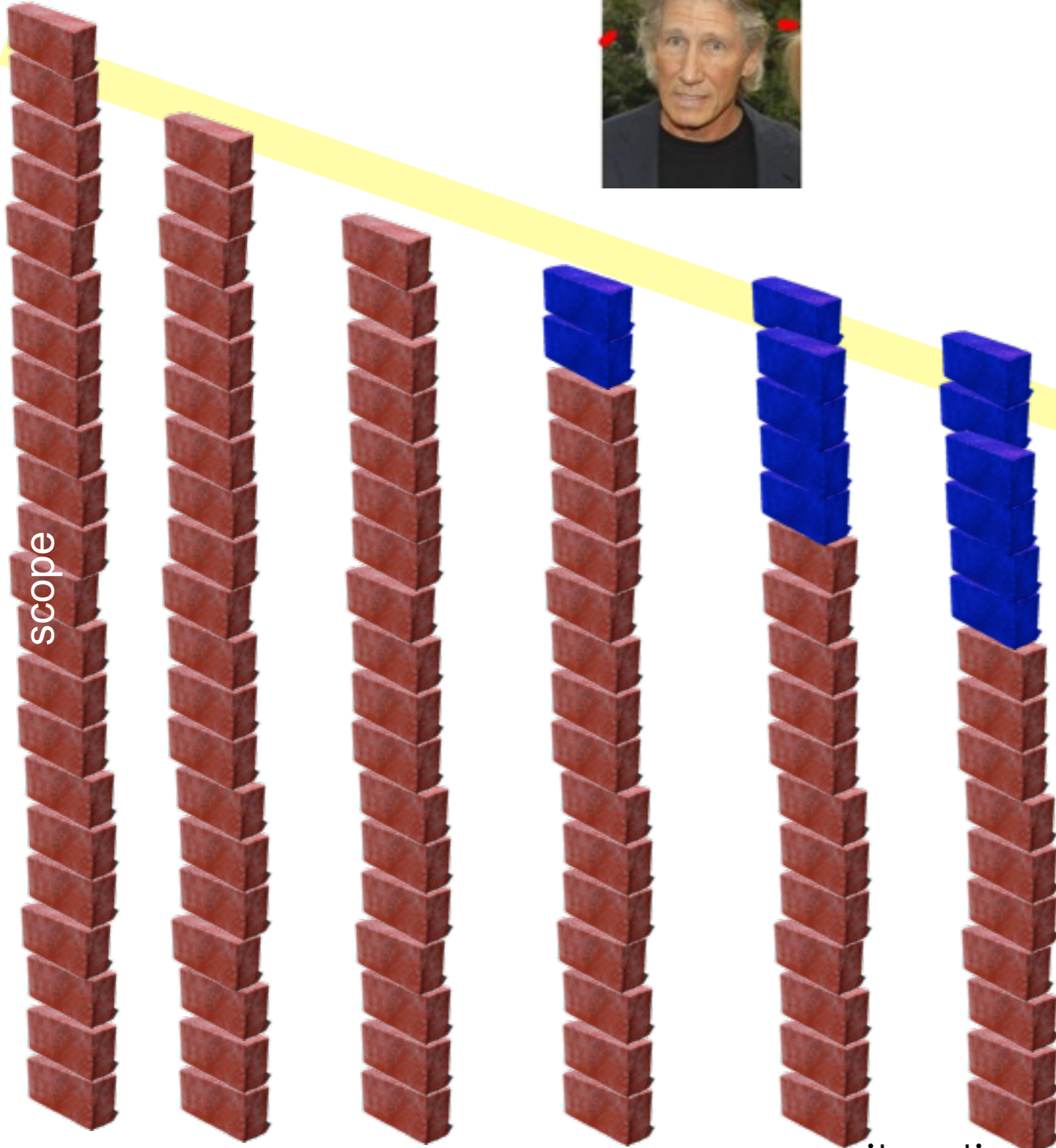
7

8

iterations



*“You gotta be kidding!”*



***“You gotta be kidding!”***

We  
are so  
screwed

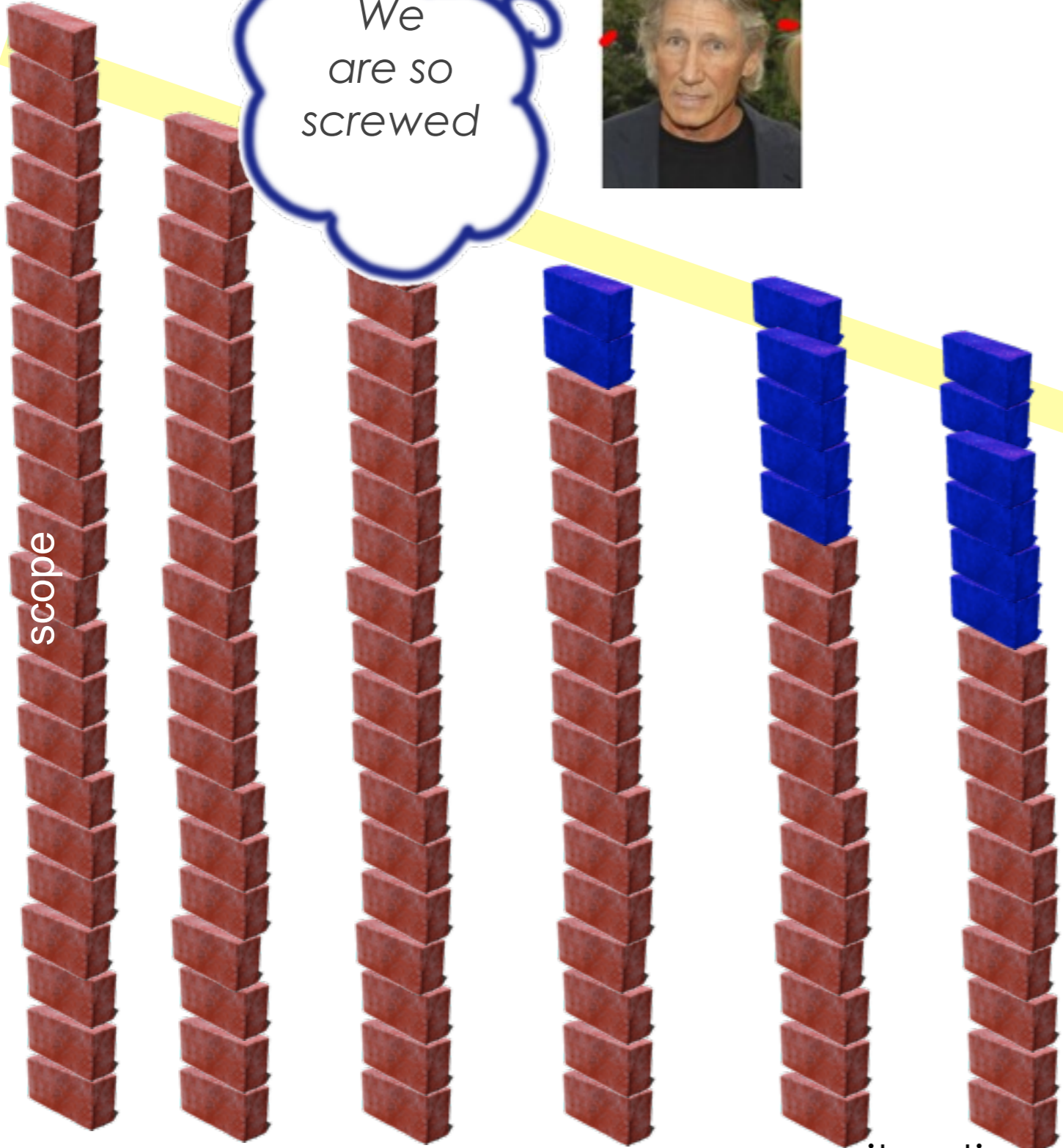


*“No Rodger, that’s the  
real power of XP – you  
can change your mind  
at any time!”*



***Isn’t it great?”***

(Melanie is good with people.)



iterations →

1

2

3

4

5

6

7

8



Roger's made a common mistake  
in XP and Agile approaches

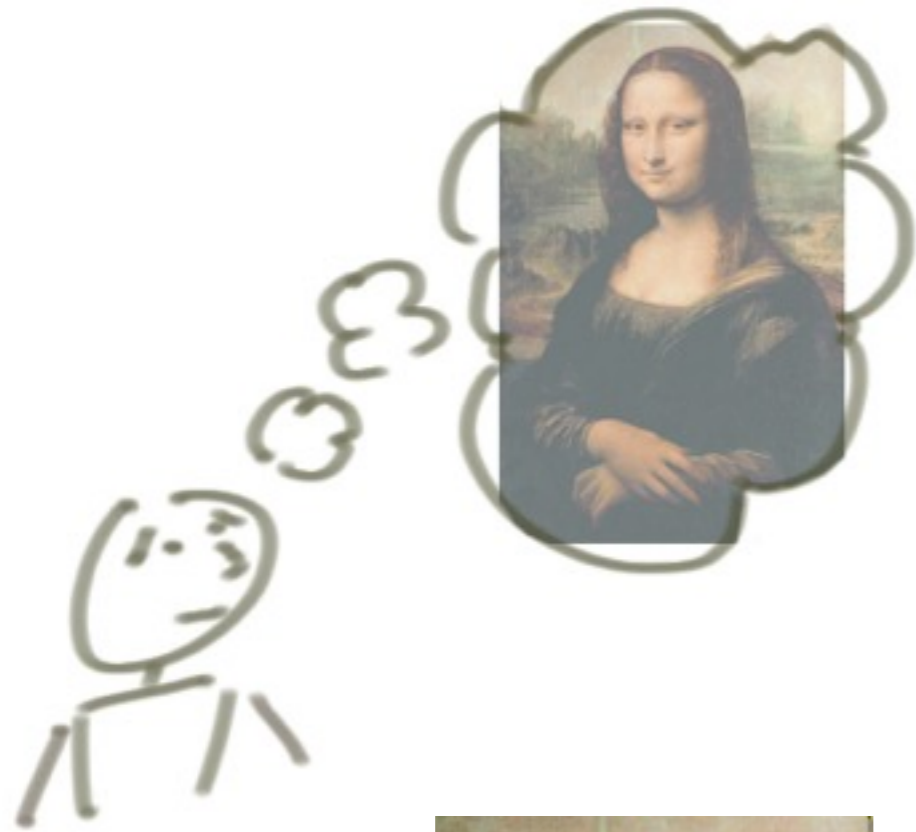
Roger's made a common mistake  
in XP and Agile approaches

He's forgotten what **iterate** means

“incrementing” builds a bit at a time



“incrementing” builds a bit at a time



1



2

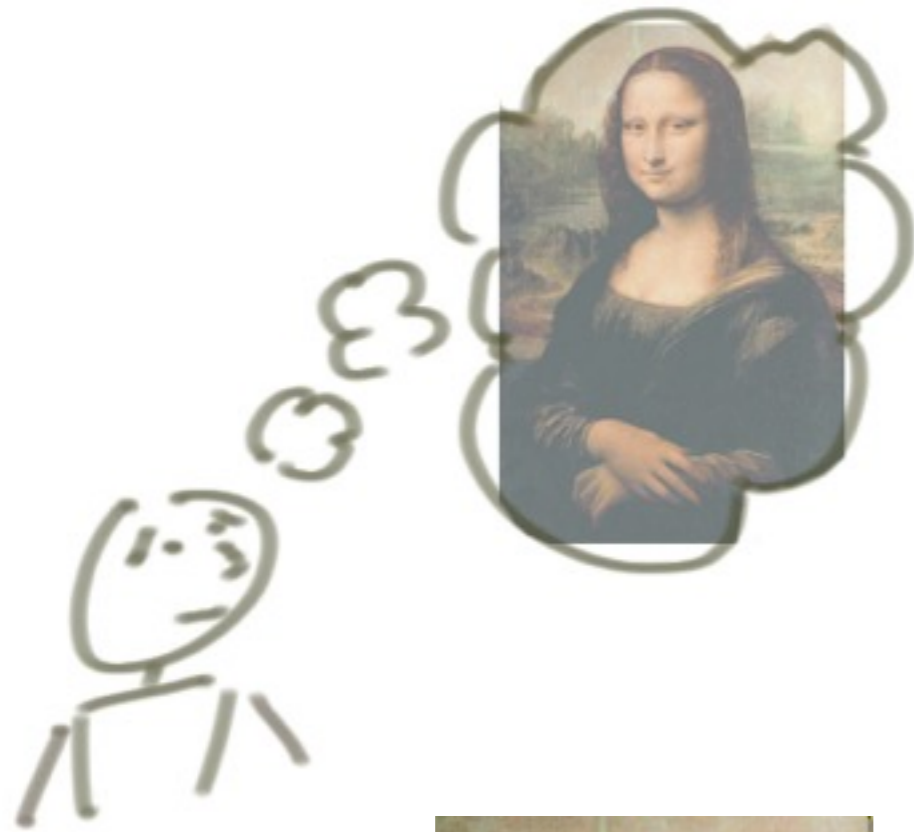


3



“incrementing” builds a bit at a time

But, incrementing  
calls for a fully  
formed idea



1



2



3





# Fred & the Werewolf



Fred Brooks,  
author of "No Silver Bullet"

# Fred & the Werewolf



Fred Brooks,  
author of "No Silver Bullet"



Steve

# Fred & the Werewolf

*“The hardest single part of building a software system is deciding precisely what to build.”*



Fred Brooks,  
author of “No Silver Bullet”



Steve

# Fred & the Werewolf

*“The hardest single part of building a software system is deciding precisely what to build.”*

Grrr....



Fred Brooks,  
author of “No Silver Bullet”



Steve

“iterating” builds a rough version, validates it, then slowly builds up quality



“iterating” builds a rough version, validates it, then slowly builds up quality



1



2



3



“iterating” builds a rough version, validates it, then slowly builds up quality

Iterating allows you to move from vague idea to realization



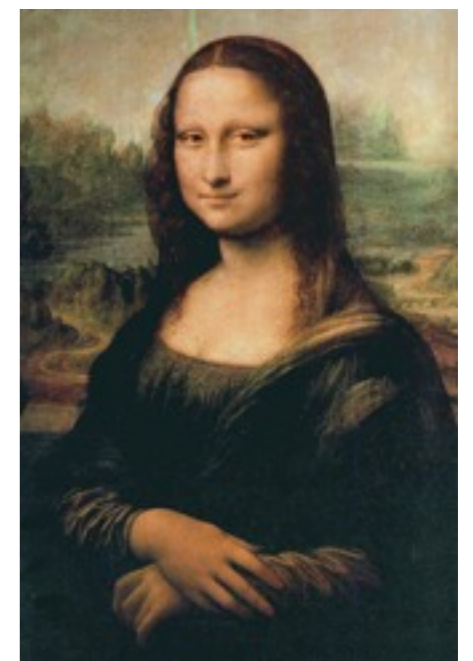
1



2



3



It's not iteration  
if you only do it once.

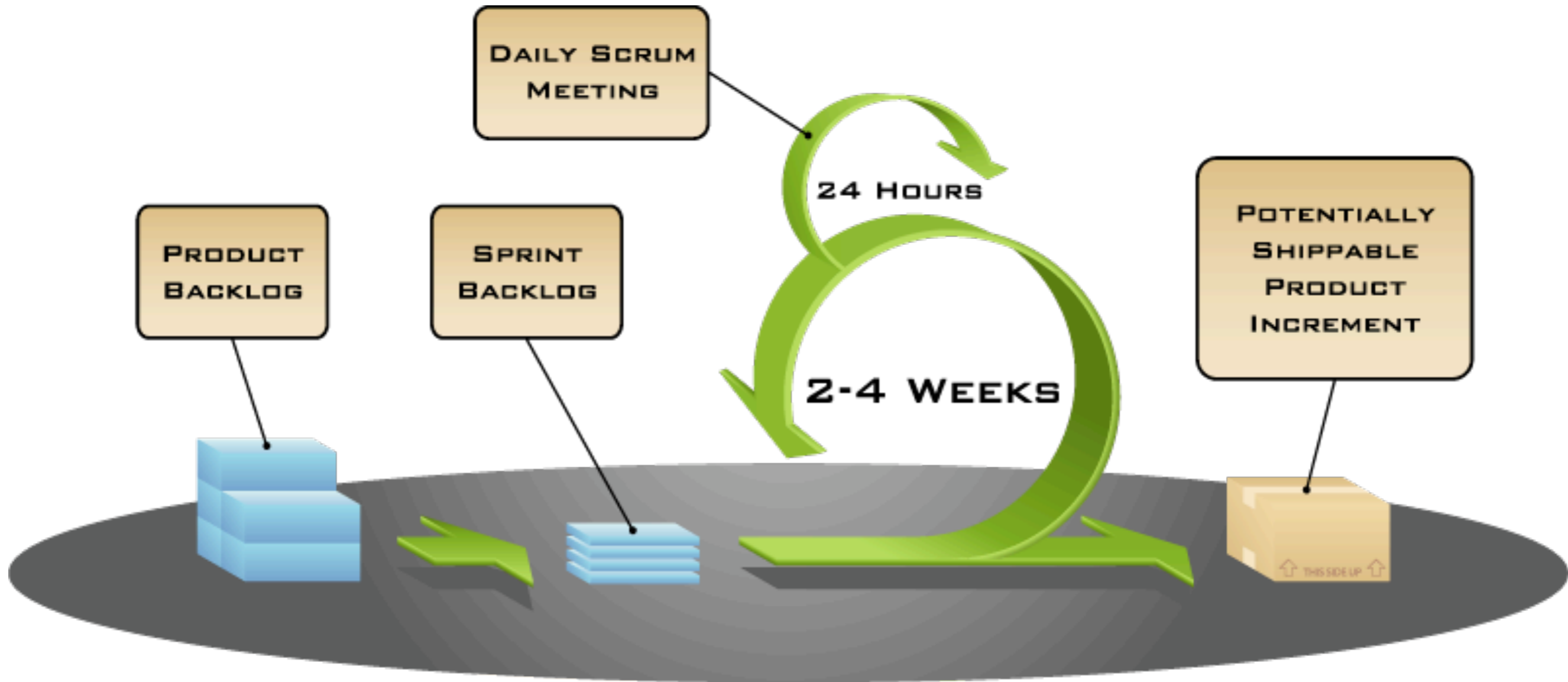
Hell



But, why would Roger not  
understand that?



# Roger saw this model...

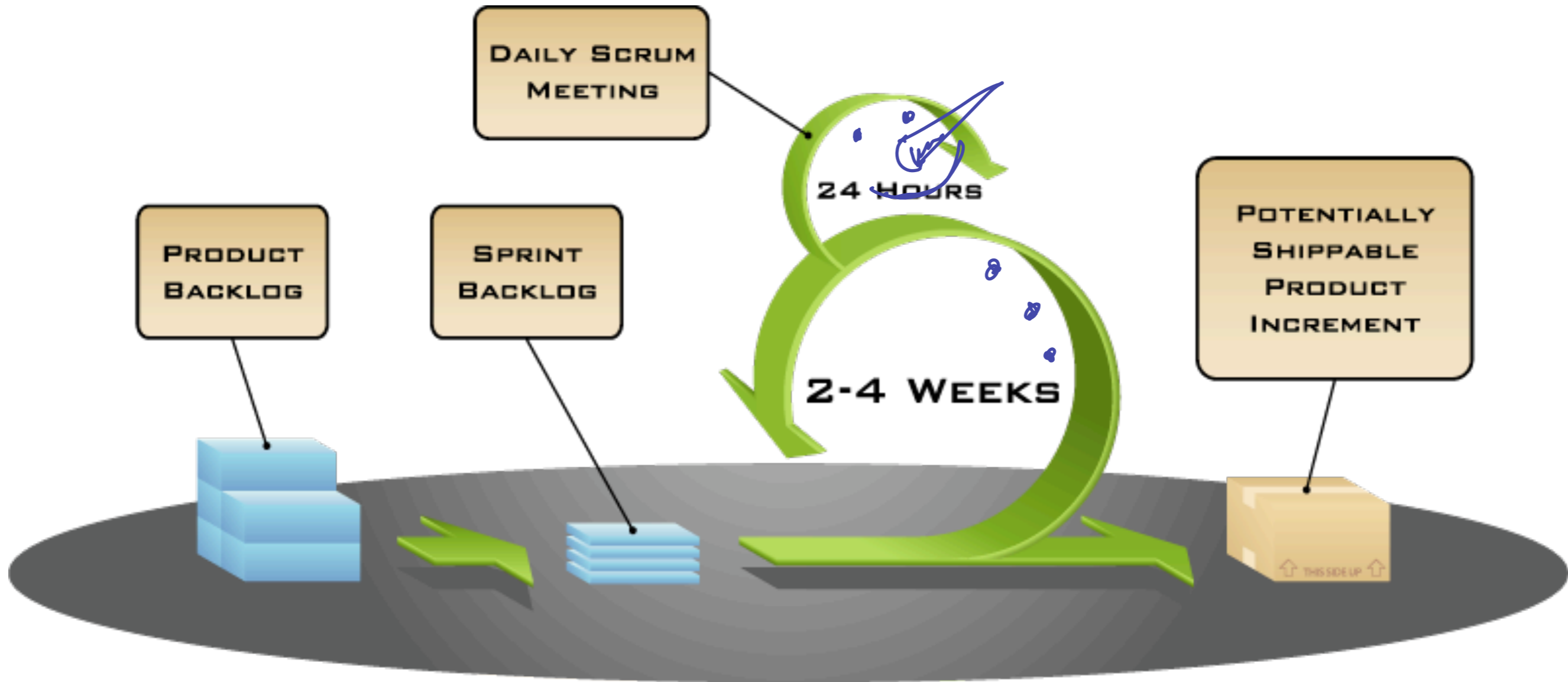


COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

## It's called "the snowman model"

(see the snowman?)

# Roger saw this model...



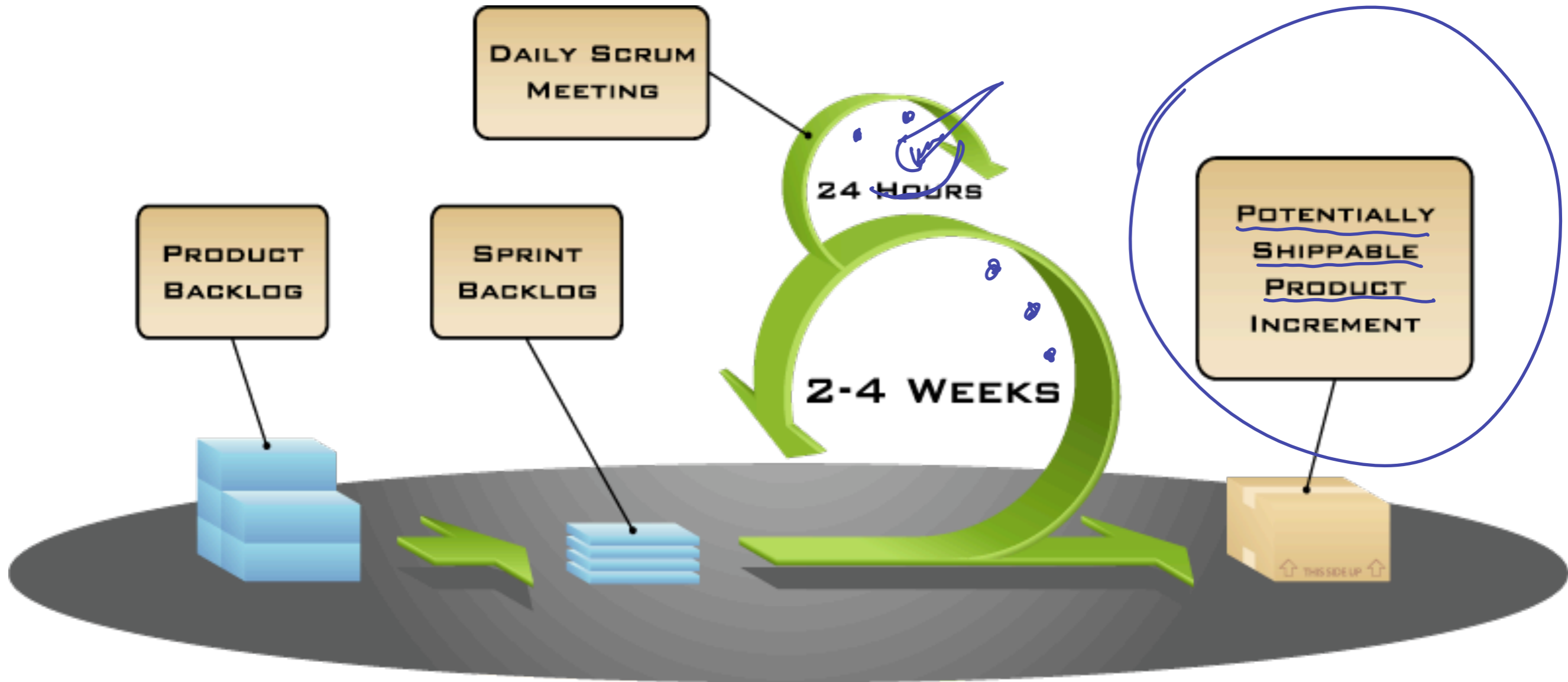
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

## It's called "the snowman model"

(see the snowman?)



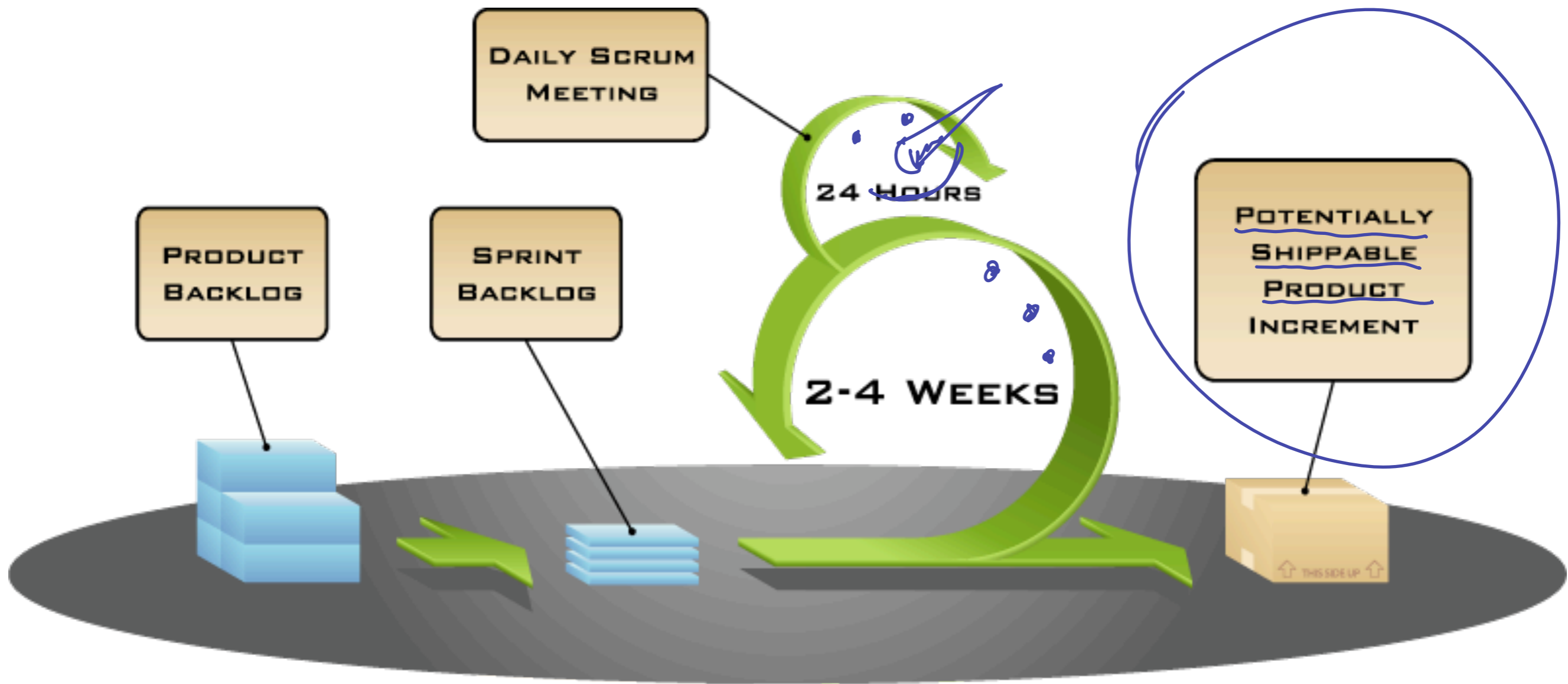
# Roger saw this model...



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

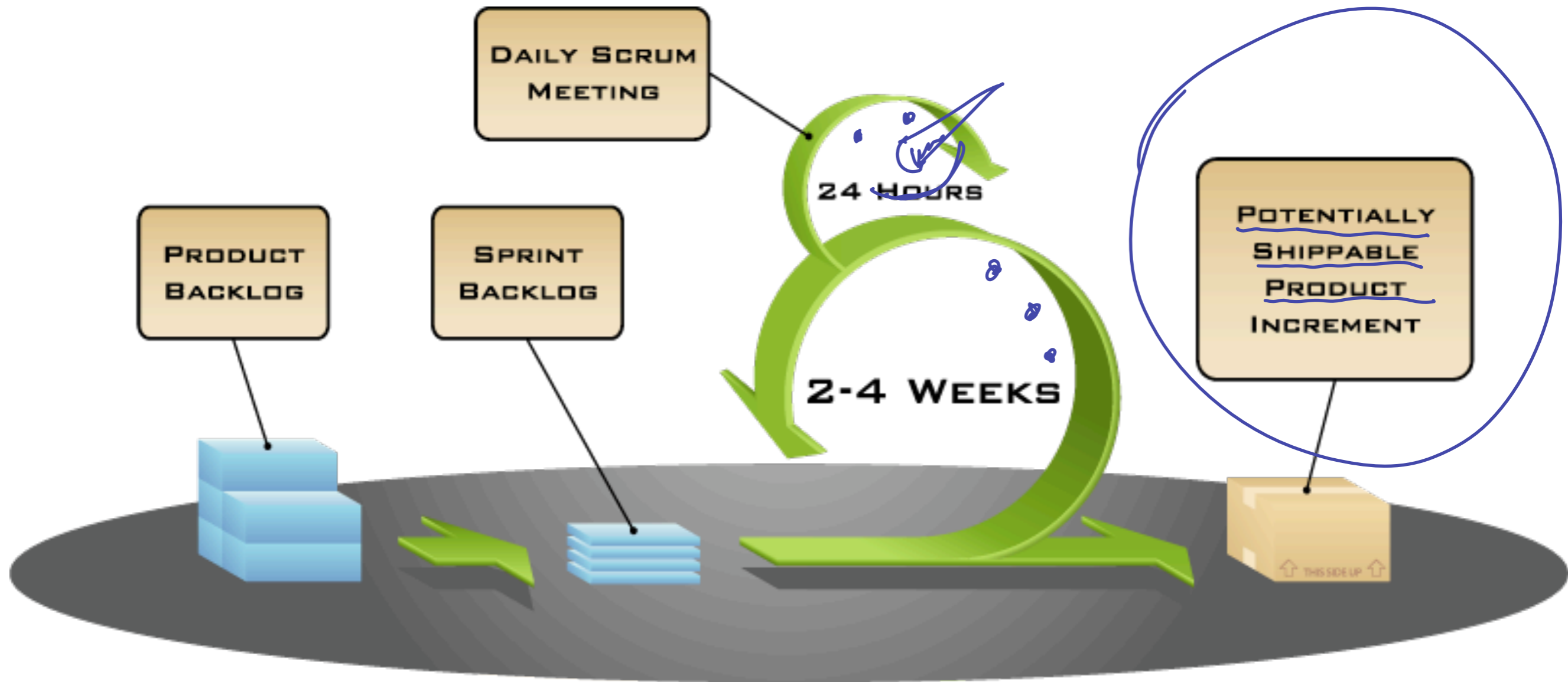
## It's called "the snowman model"

(see the snowman?)



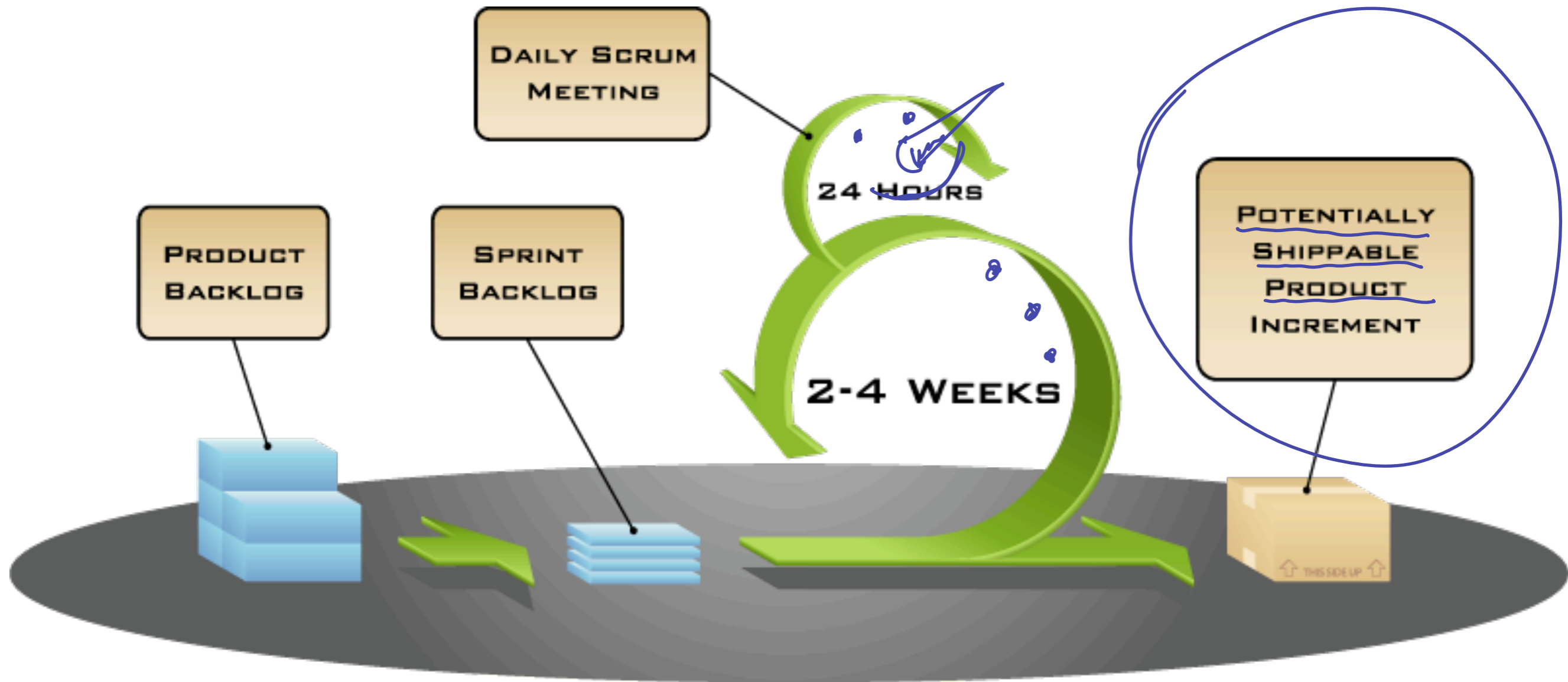
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

# It can't be half-baked if you're going to ship it... right?

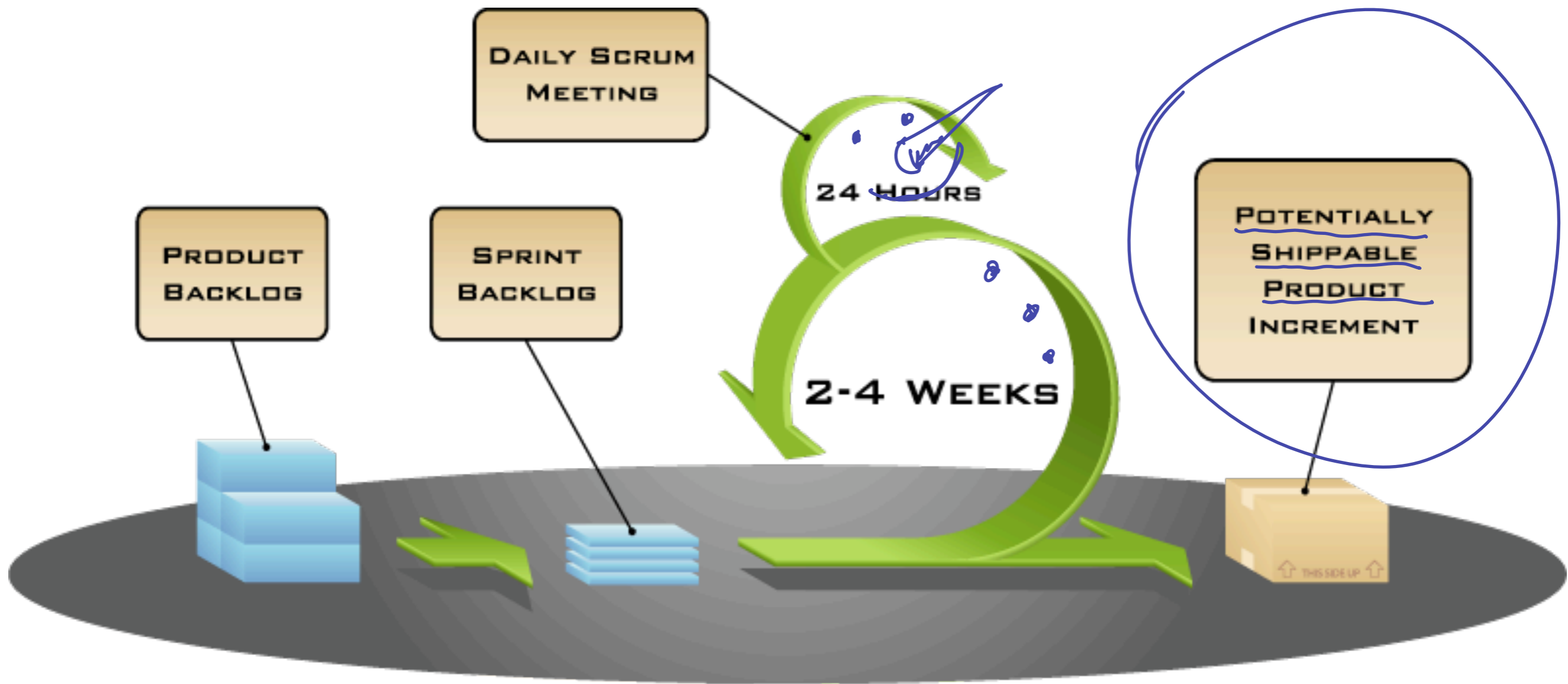


COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

# It can't be half-baked if you're going to ship it... right?

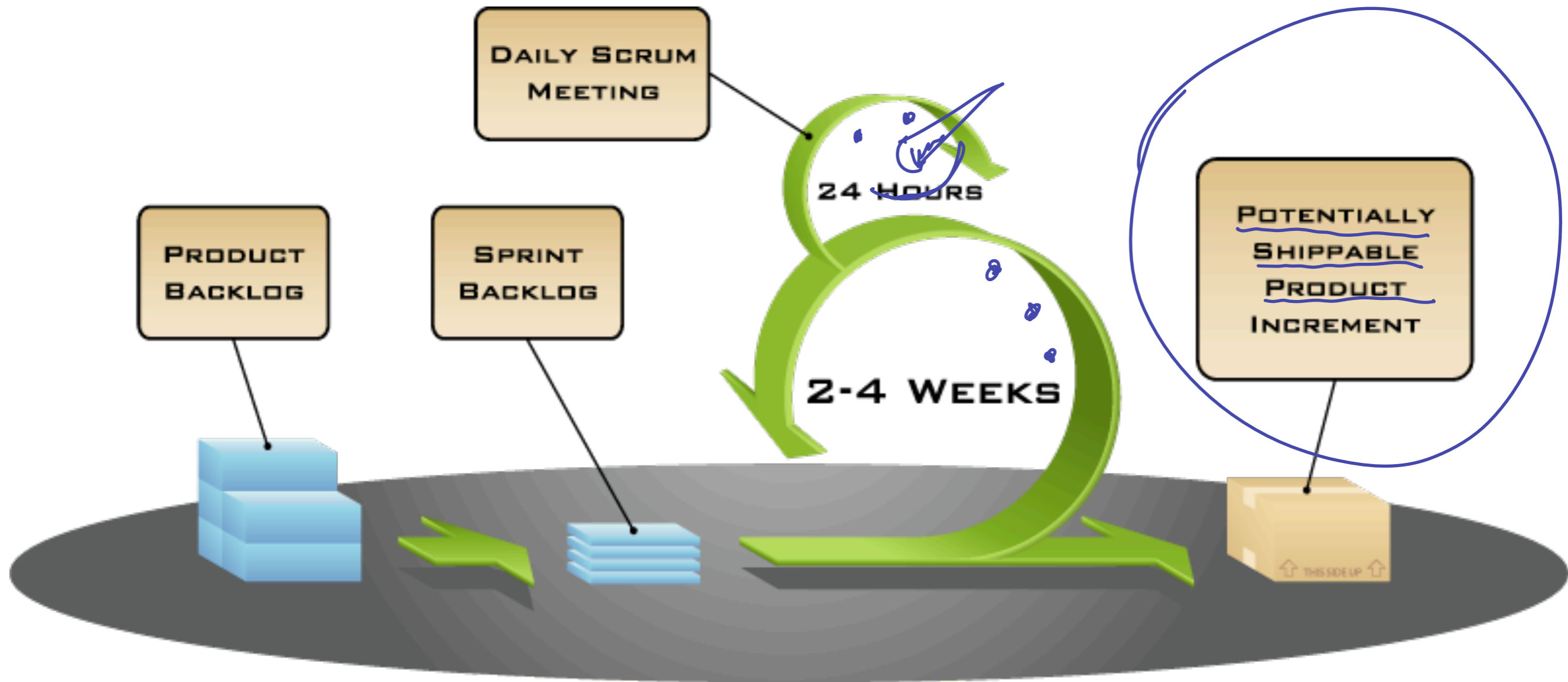


COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

# But, it can if you intend to iterate.



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE



It's not Rogers fault he didn't think  
about iteration...



It's not Rogers fault he didn't think  
about iteration...

the snowman is missing a couple  
balls











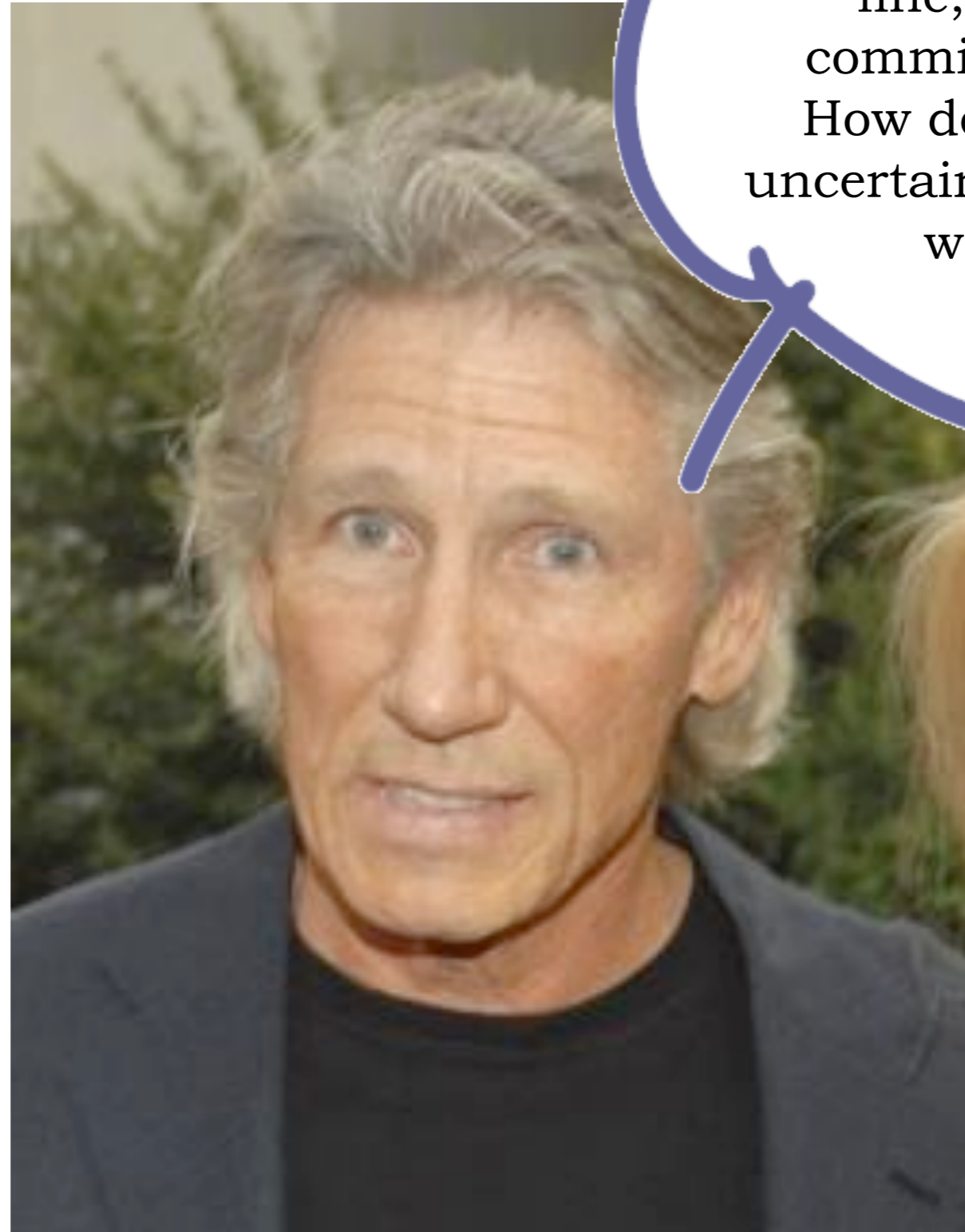












Look, that iteration stuff is fine, but we've got commitments to keep. How do I deal with the uncertainty of not knowing what I'll get?



Roger's got a point

Roger's got a point

Software is often a line item in a larger plan. Failing a release date may put that bigger plan at risk

Roger's got a point

Software is often a line item in a larger plan. Failing a release date may put that bigger plan at risk

(Failing the release date may put Roger's career at risk)





It seems logical that we need to know what we want in order to estimate its construction and get it on schedule, right?

It seems logical that we need to know what we want in order to estimate its construction and get it on schedule, right?

But we know that we can't *really* know...



Planning for iteration might have helped Roger, but that still doesn't solve Roger's fear of uncertainty



Here's three strategies that might  
help



This is **John**







This is **John**

John is somewhat single  
minded.



This is **John**

John is somewhat single minded.

He focuses on **business value...**



This is **John**

John is somewhat single minded.

He focuses on **business value...**  
well sort of.



This is **John**

John is somewhat single minded.

He focuses on **business value...**

well sort of.

John's Strategy: **Follow the Money**





John follows user stories back to their source

# User Stories Build Software





John follows user stories back to their source

# User Constituencies

(The people that will use some solution to meet business goals)



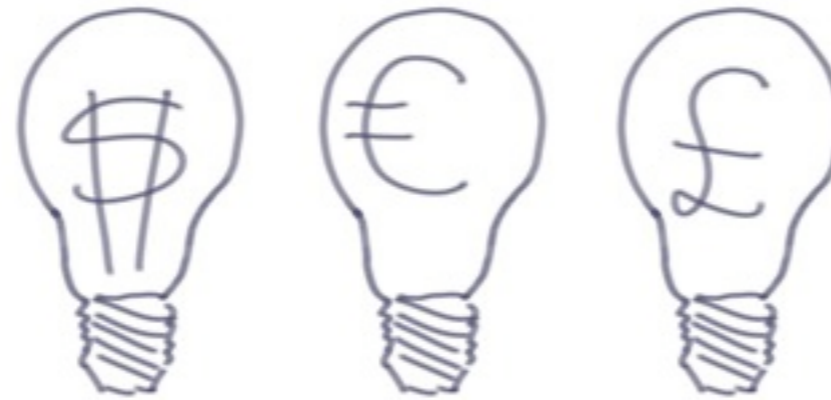
# User Stories Build Software





# Business Goals

(Increase Revenue, Reduce Costs)



John follows user stories back to their source

# User Constituencies

(The people that will use some solution to meet business goals)



# User Stories Build Software



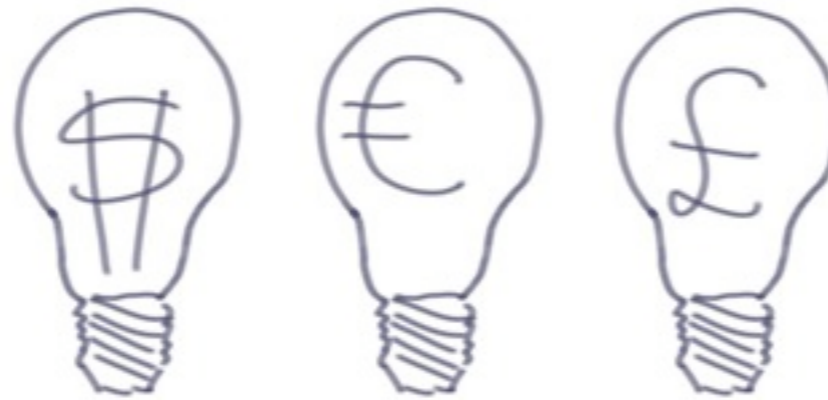


Prioritize before you prioritize



## Business Goals

(Increase Revenue,  
Reduce Costs)



## User Constituencies

(The people that will  
use some solution to  
meet business goals)



## User Stories Build Software

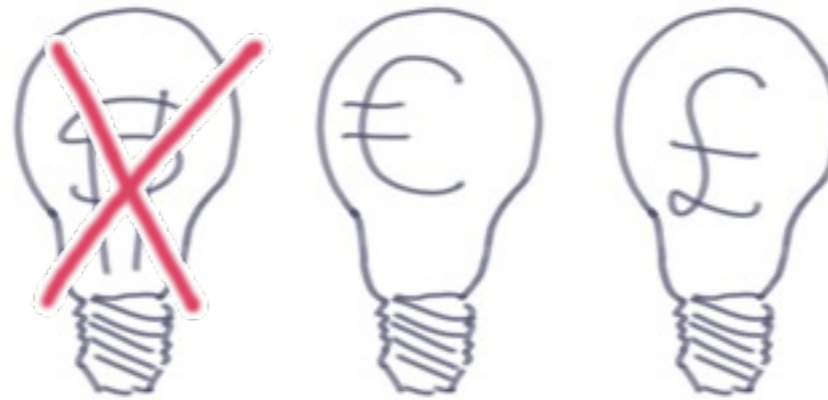


Prioritize before you prioritize



## Business Goals

(Increase Revenue, Reduce Costs)



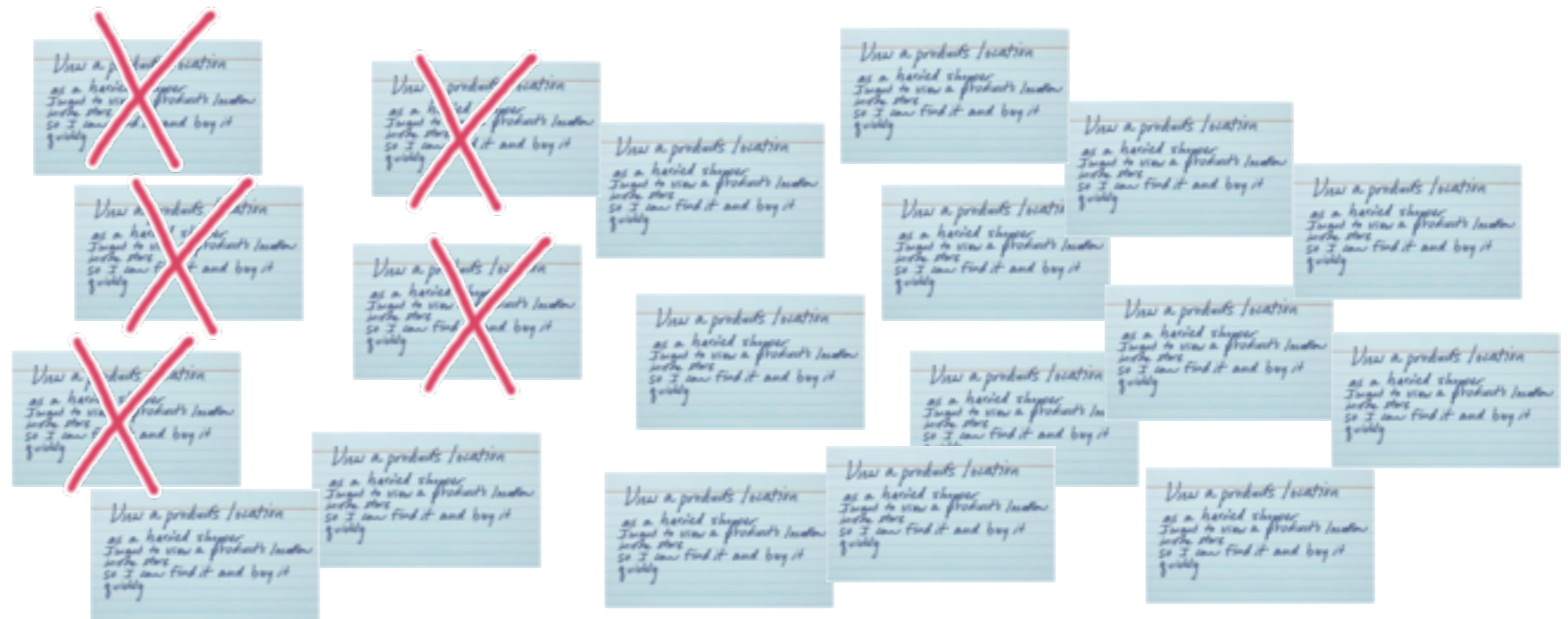
John knows he can get money faster by prioritizing goals first

## User Constituencies

(The people that will use some solution to meet business goals)



## User Stories Build Software



Prioritize before you prioritize



## Business Goals

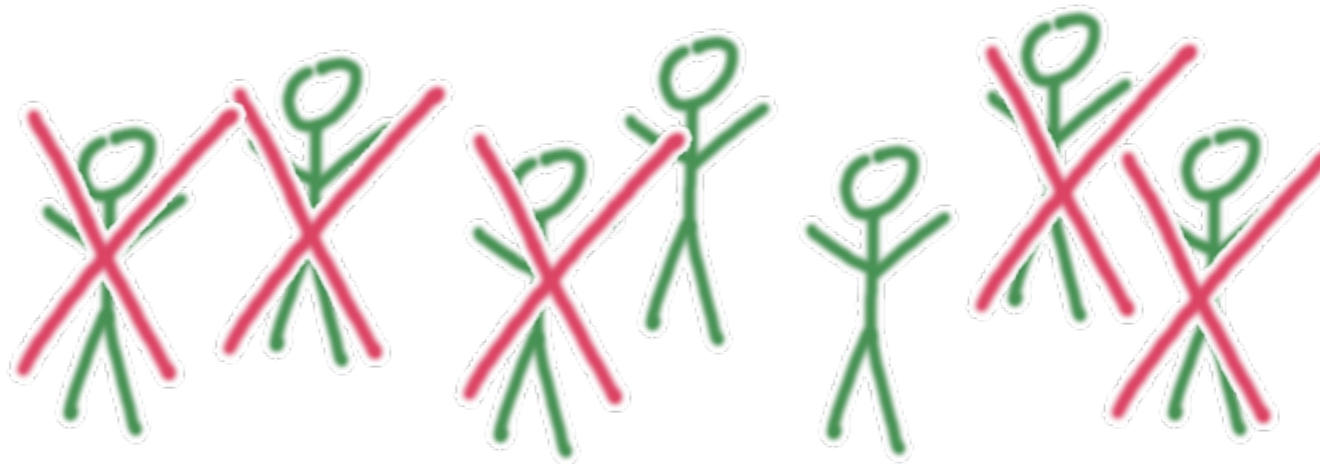
(Increase Revenue, Reduce Costs)



John knows he can get money faster by prioritizing goals first

## User Constituencies

(The people that will use some solution to meet business goals)



## User Stories Build Software



Prioritize before you prioritize



## Business Goals

(Increase Revenue,  
Reduce Costs)



## User Constituencies

(The people that will  
use some solution to  
meet business goals)



## User Stories Build Software



Prioritize before you prioritize

## Business Goals

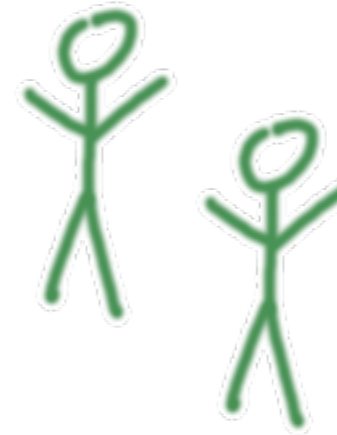
(Increase Revenue,  
Reduce Costs)



“Fewer goals =  
less software”

## User Constituencies

(The people that will  
use some solution to  
meet business goals)



## User Stories Build Software





This is **Paul**





## This is **Paul**

(It's an old picture. He's not looking quite as good these days)





This is **Paul**

(It's an old picture. He's not looking quite as good these days)

Paul has a problem



This is **Paul**

(It's an old picture. He's not looking quite as good these days)

Paul has a problem

“As a **frustrated boyfriend,**



This is **Paul**

(It's an old picture. He's not looking quite as good these days)

Paul has a problem

“As a **frustrated boyfriend**,  
I want to **leave my lover**”



## This is **Paul**

(It's an old picture. He's not looking quite as good these days)

Paul has a problem

“As a **frustrated boyfriend**,  
I want to **leave my lover**  
so that **I can be happier with**  
**someone else.**”



## This is **Paul**

(It's an old picture. He's not looking quite as good these days)

Paul has a problem

“As a **frustrated boyfriend**,  
I want to **leave my lover**  
so that **I can be happier with**  
**someone else.**”

“I've got a lot of options.”



This is **Paul**

(It's an old picture. He's not looking quite as good these days)

Paul has a problem

“As a **frustrated boyfriend**, I want to **leave my lover** so that **I can be happier with someone else.**”

“I've got a lot of options.”

Paul's Strategy: **Don't choose your solution too early**





Paul defers writing user stories that describe the software till the last responsible moment.





Paul defers writing user stories that describe the software till the last responsible moment.

Instead, he writes users stories about the users and what they need to accomplish.



Paul defers writing user stories that describe the software till the last responsible moment.

Instead, he writes users stories about the users and what they need to accomplish.

When working with Melanie to estimate, he discusses all the ways – sometimes as many as 50 – that the user can satisfy their goals.



This is **Pete** and **Roger**



## This is **Pete and Roger**

(It's a different Roger - He looks a little less dazed than the other Roger)



## This is **Pete and Roger**

(It's a different Roger - He looks a little less dazed than the other Roger)

They have a problem. What they want may cost more than they can afford.





## This is **Pete and Roger**

(It's a different Roger - He looks a little less dazed than the other Roger)

They have a problem. What they want may cost more than they can afford.

But, they know how to vary and build up quality to stay under budget, but maximize value.



## This is **Pete and Roger**

(It's a different Roger - He looks a little less dazed than the other Roger)

They have a problem. What they want may cost more than they can afford.

But, they know how to vary and build up quality to stay under budget, but maximize value.

# Pete & Roger's Strategy: **Build up feature quality iteration by iteration**





Pete & Roger prepare a backlog for their bus



engine  
transmission  
brakes  
suspension  
seats  
steering wheel  
beer cooler  
...

# Pete & Roger prepare a backlog for their bus



He's a real doofus.

engine  
transmission  
brakes  
suspension  
seats  
steering wheel  
beer cooler  
...

Hey – you need to prioritize those!





Pete & Roger prepare a backlog for their bus

They know they need **all** the features

engine  
transmission  
brakes  
suspension  
seats  
steering wheel  
beer cooler  
...



Pete & Roger prepare a backlog for their bus

They know they need **all** the features

But they know that all buses don't cost the same

Each essential feature varies in quality affecting the final cost

engine  
transmission  
brakes  
suspension  
seats  
steering wheel  
beer cooler  
...



low cost



moderate cost



high cost



Pete and Roger have a handy heuristic for slicing up quality

Pete and Roger have a handy heuristic for slicing up quality

**Necessity:** what minimal characteristics are necessary for this feature?



Pete and Roger have a handy heuristic for slicing up quality

**Necessity:** what minimal characteristics are necessary for this feature?

**Flexibility:** what would make this feature useful in more situations?

Pete and Roger have a handy heuristic for slicing up quality

**Necessity:** what minimal characteristics are necessary for this feature?

**Flexibility:** what would make this feature useful in more situations?

**Safety:** what would make this feature safer for me to use?

Pete and Roger have a handy heuristic for slicing up quality

**Necessity:** what minimal characteristics are necessary for this feature?

**Flexibility:** what would make this feature useful in more situations?

**Safety:** what would make this feature safer for me to use?

**Comfort, Luxury, and Performance:** what would make this feature more desirable to use?





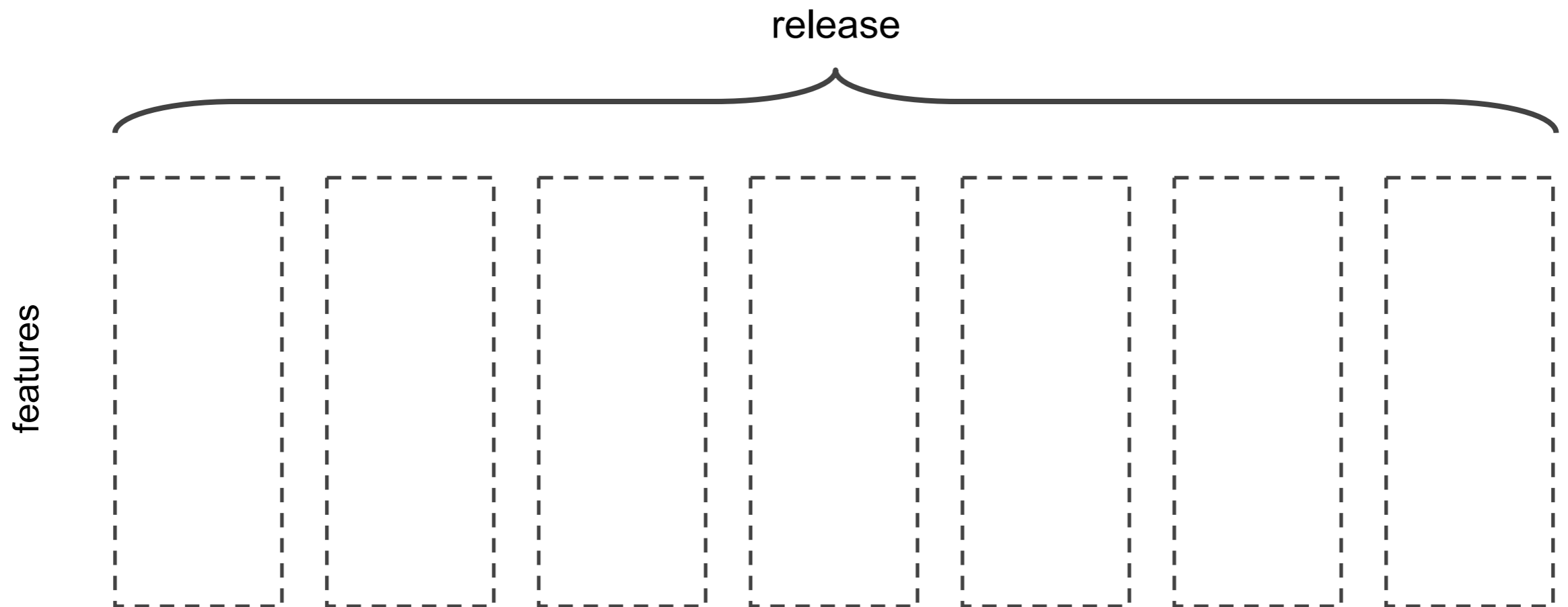
Pete and Roger have learned the hard way that building each story to an ideal quality level is risky.

(Although she has her qualities, Melanie isn't as good at estimation as you might think.)



Pete and Roger have learned the hard way that building each story to an ideal quality level is risky.

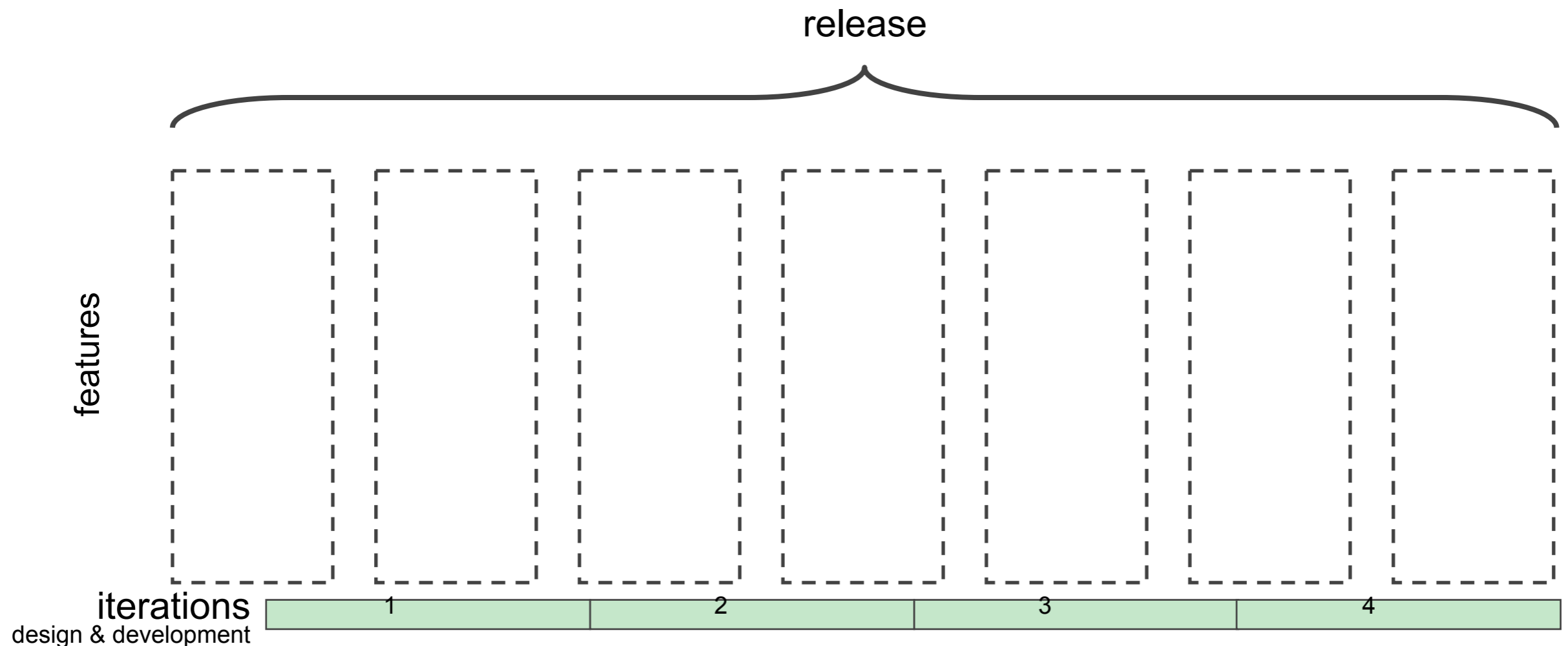
(Although she has her qualities, Melanie isn't as good at estimation as you might think.)





Pete and Roger have learned the hard way that building each story to an ideal quality level is risky.

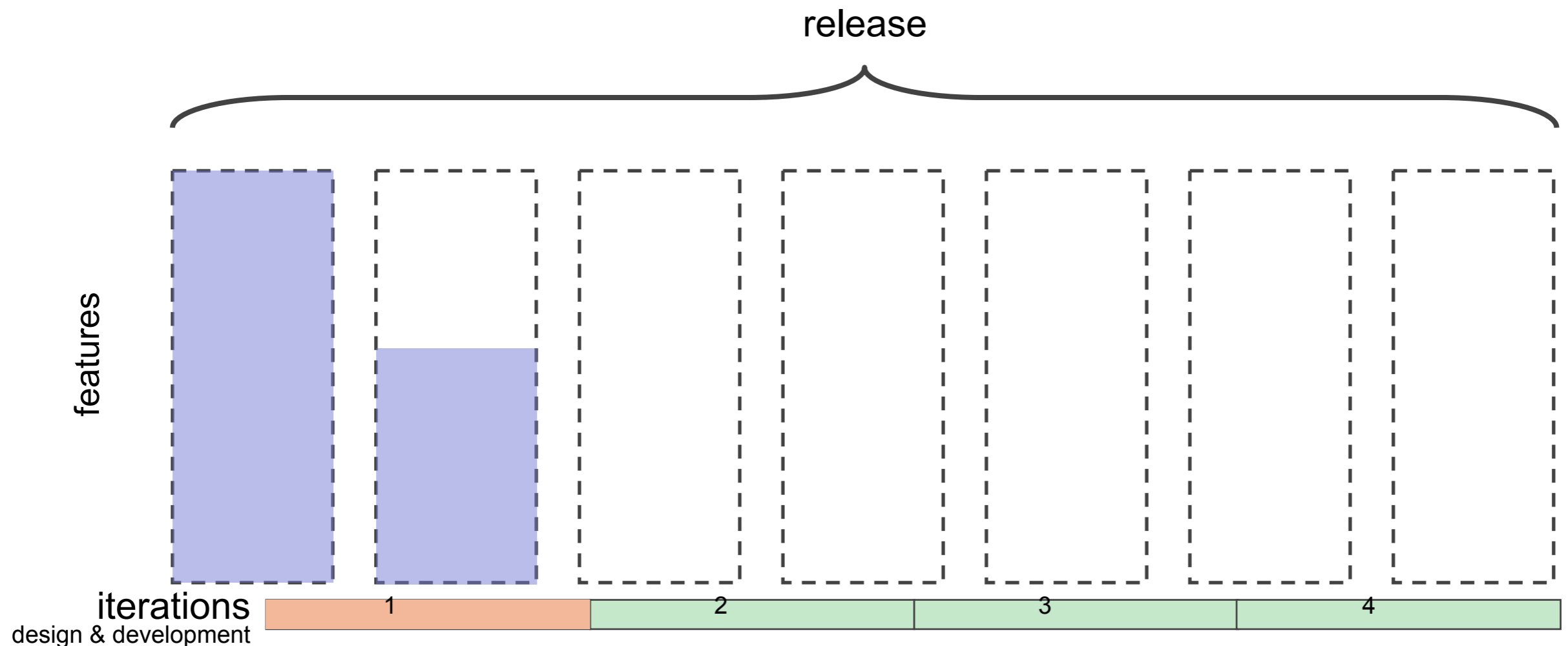
(Although she has her qualities, Melanie isn't as good at estimation as you might think.)





Pete and Roger have learned the hard way that building each story to an ideal quality level is risky.

(Although she has her qualities, Melanie isn't as good at estimation as you might think.)

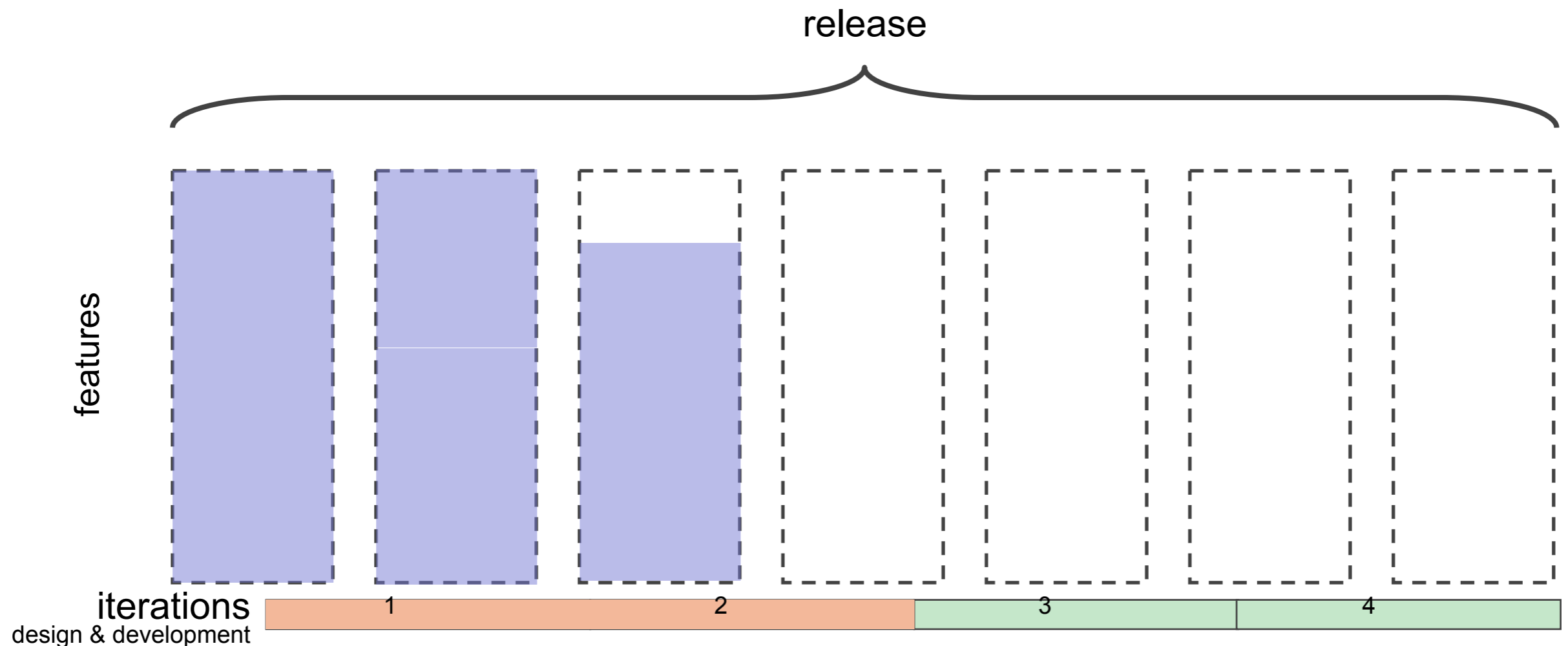






Pete and Roger have learned the hard way that building each story to an ideal quality level is risky.

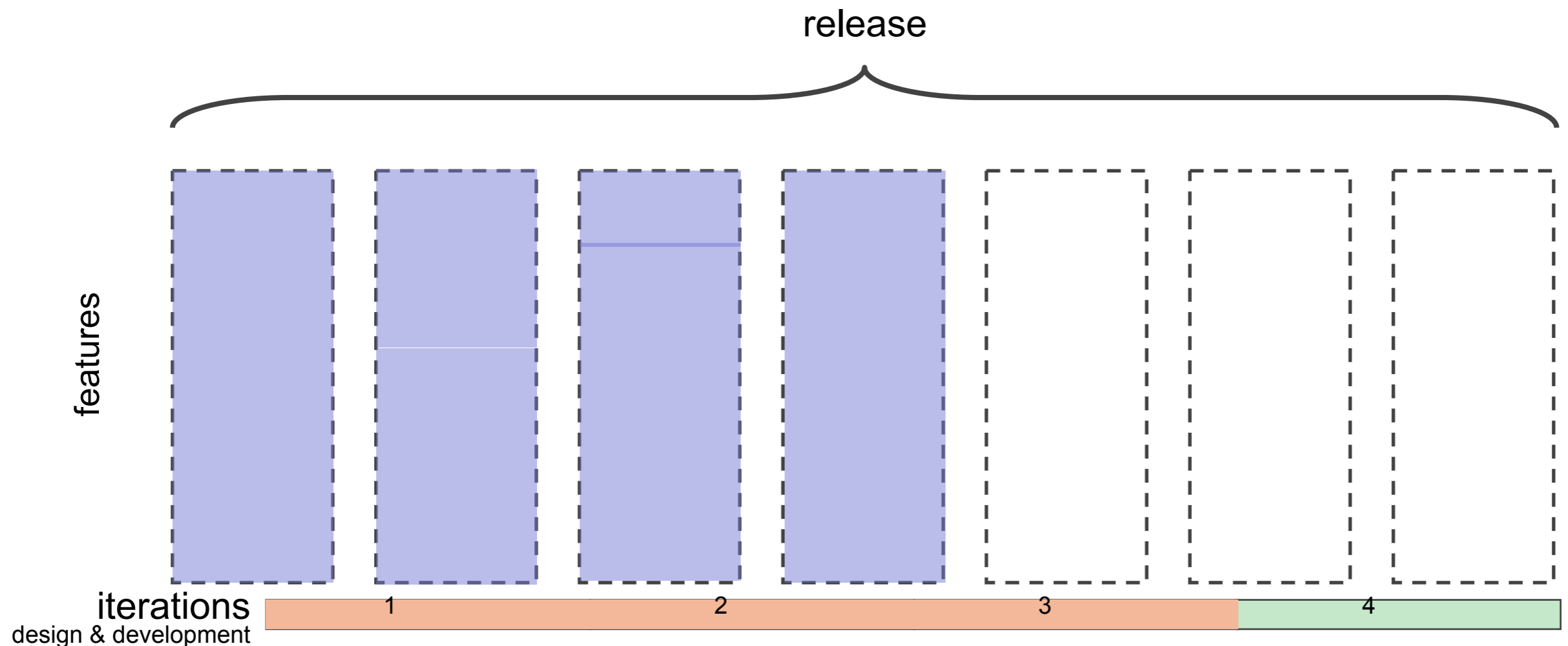
(Although she has her qualities, Melanie isn't as good at estimation as you might think.)





Pete and Roger have learned the hard way that building each story to an ideal quality level is risky.

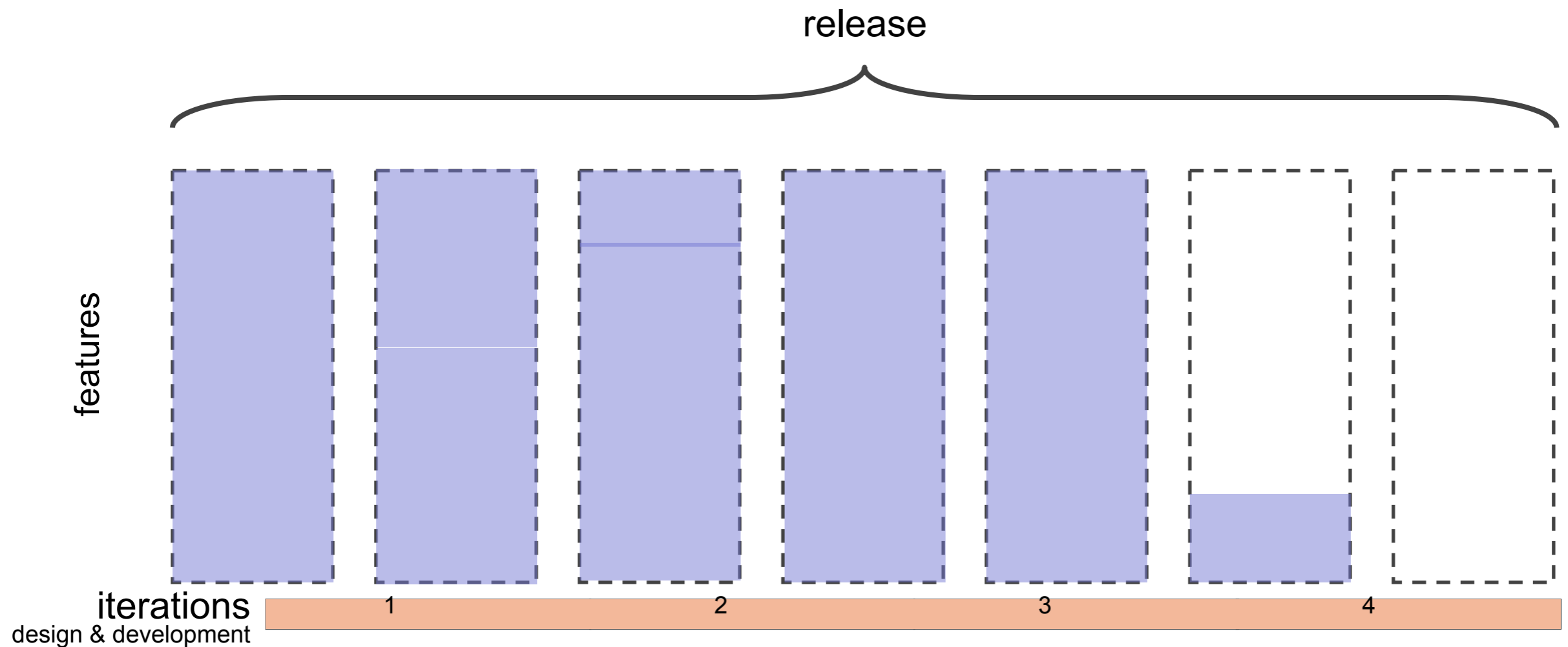
(Although she has her qualities, Melanie isn't as good at estimation as you might think.)





Pete and Roger have learned the hard way that building each story to an ideal quality level is risky.

(Although she has her qualities, Melanie isn't as good at estimation as you might think.)



# Pete and Roger leverage iteration

# Pete and Roger leverage iteration



Iterating affords building up quality over time

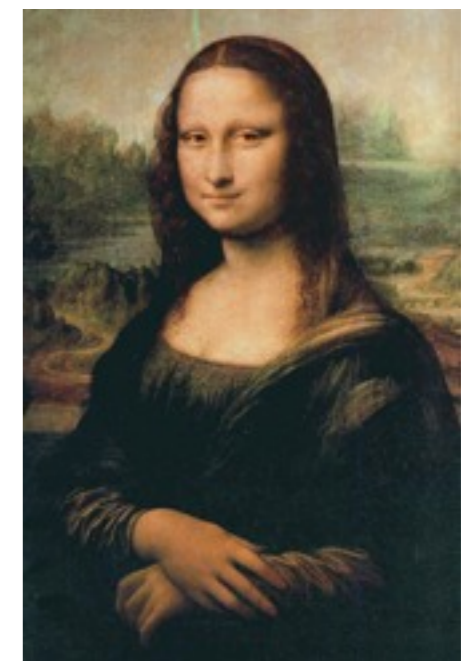
1



2



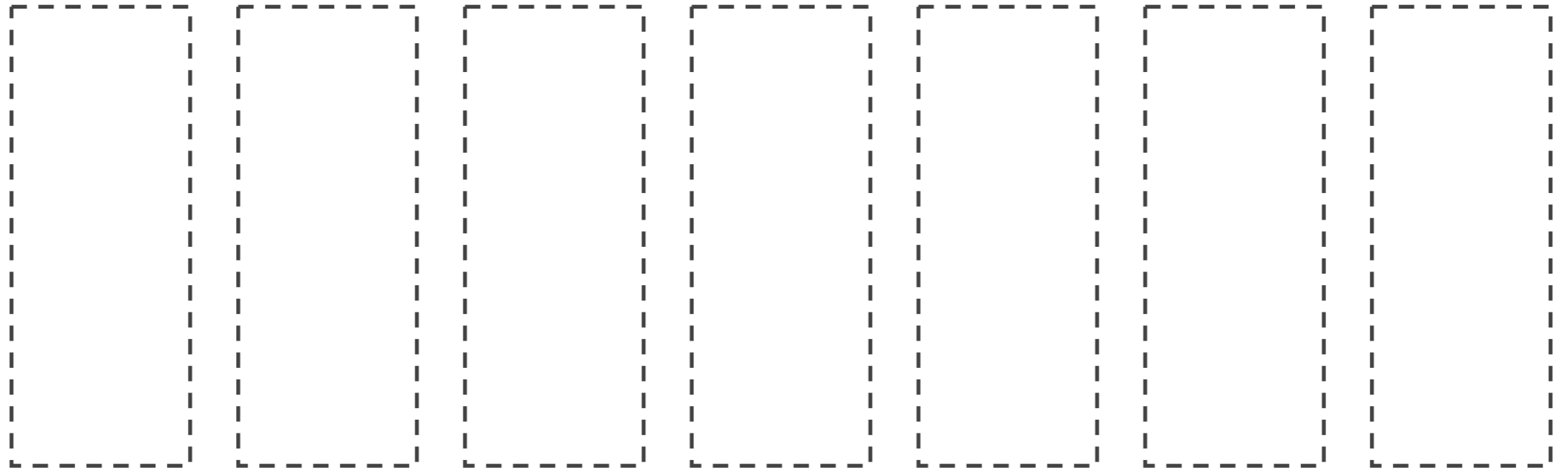
3



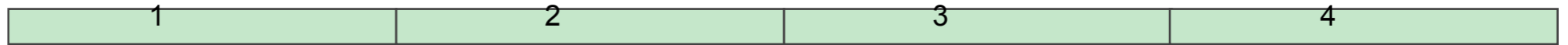
release



user tasks to support



iterations  
design & development



1

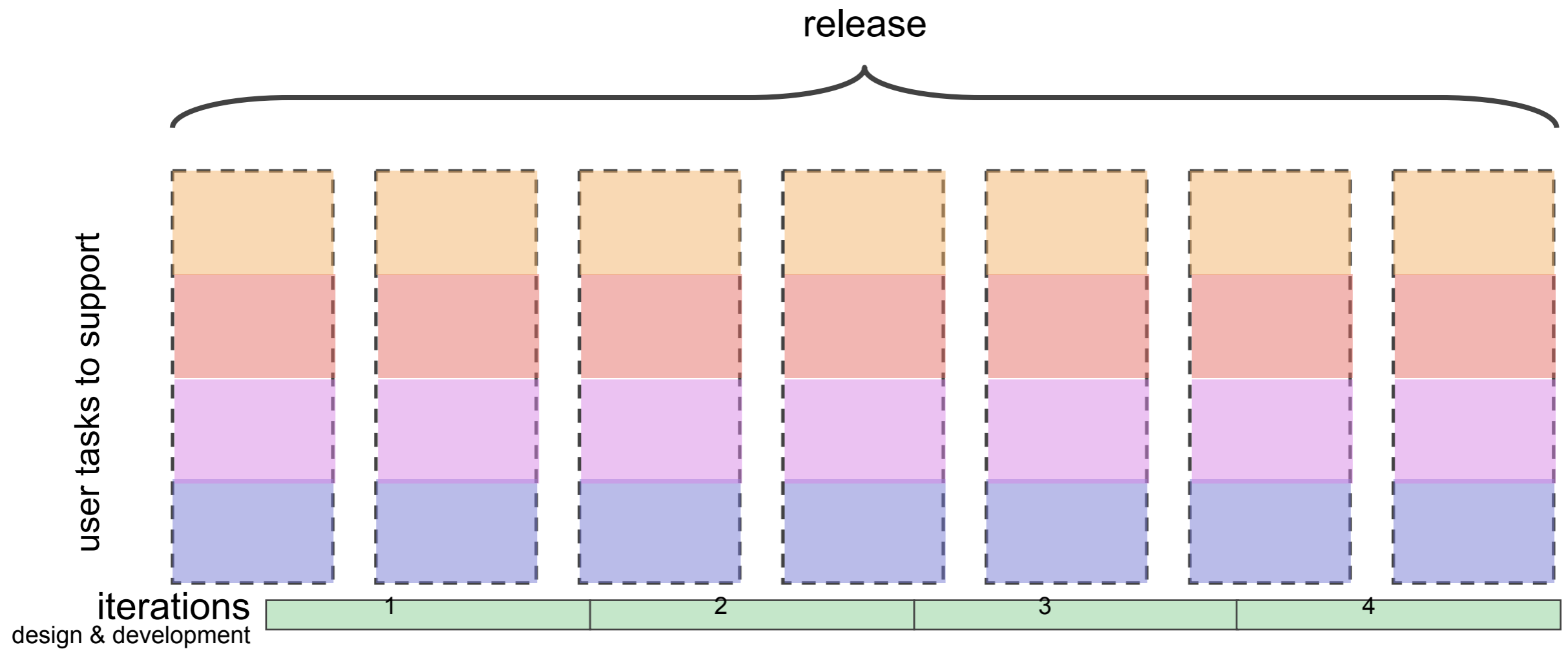
2

3

4



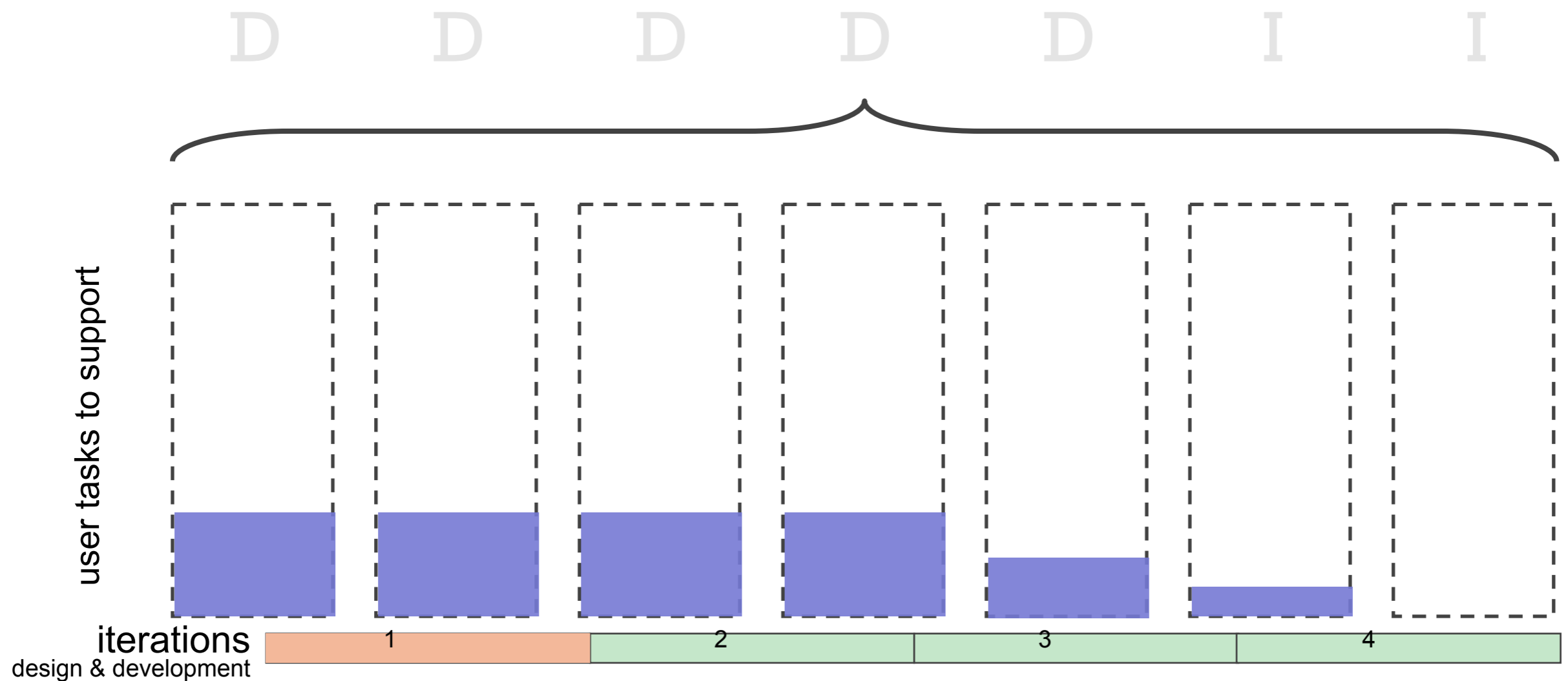
Pete and Roger know that each bus feature can be split into user stories based on quality characteristics.





Pete and Roger know that each bus feature can be split into user stories based on quality characteristics.

In early iterations Pete and Roger focus on necessity, then and flexibility and safety, then finish off with luxury



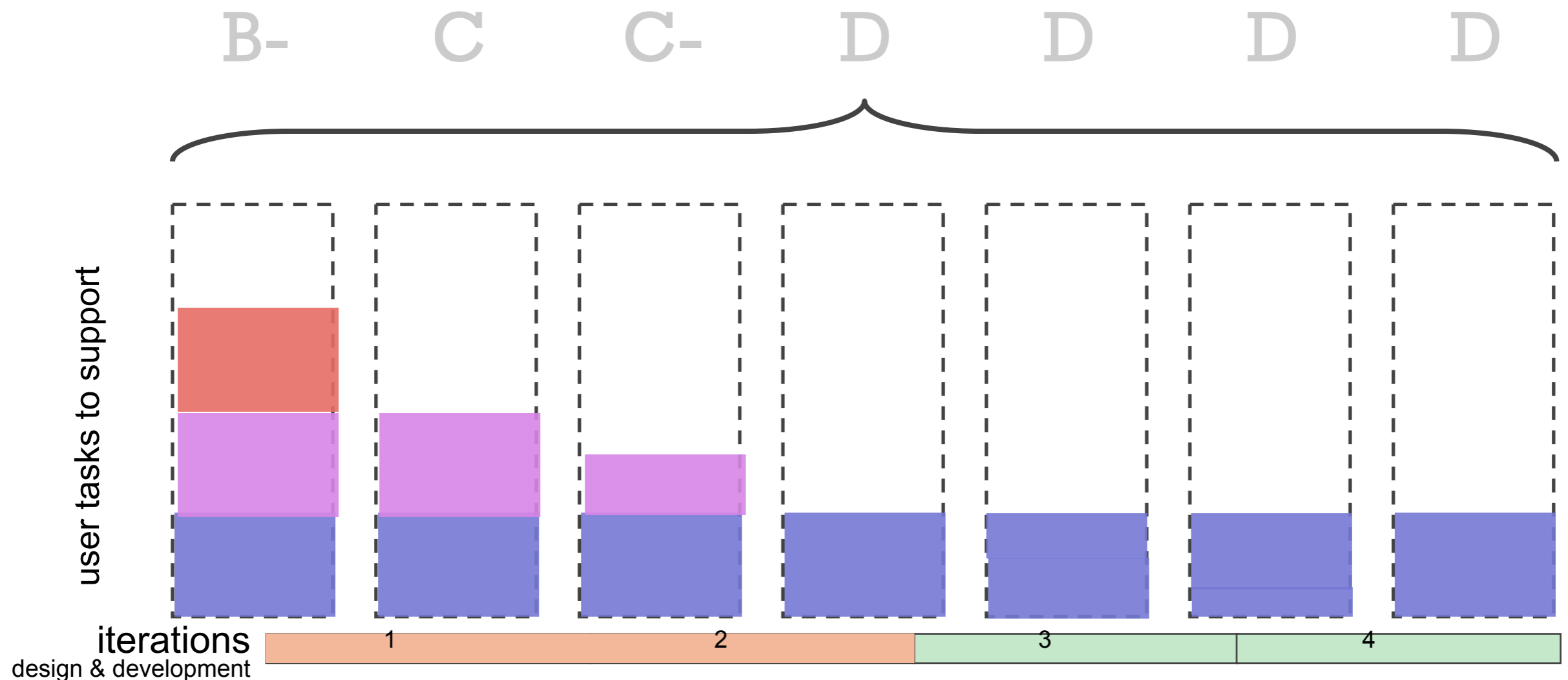




Pete and Roger know that each bus feature can be split into user stories based on quality characteristics.

In early iterations Pete and Roger focus on necessity, then and flexibility and safety, then finish off with luxury

At each iteration they give their features a quality grade, then evaluate their bus report card.

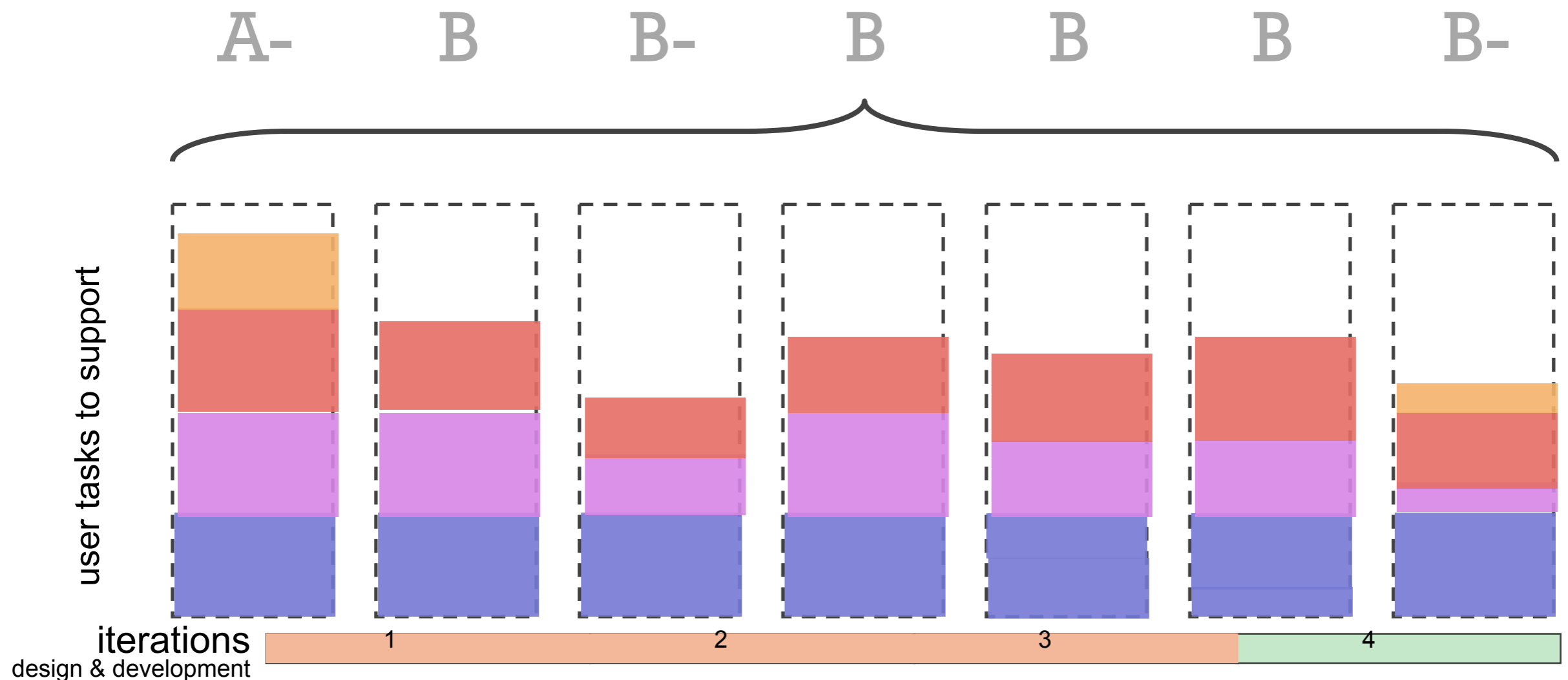




Pete and Roger know that each bus feature can be split into user stories based on quality characteristics.

In early iterations Pete and Roger focus on necessity, then and flexibility and safety, then finish off with luxury

At each iteration they give their features a quality grade, then evaluate their bus report card.

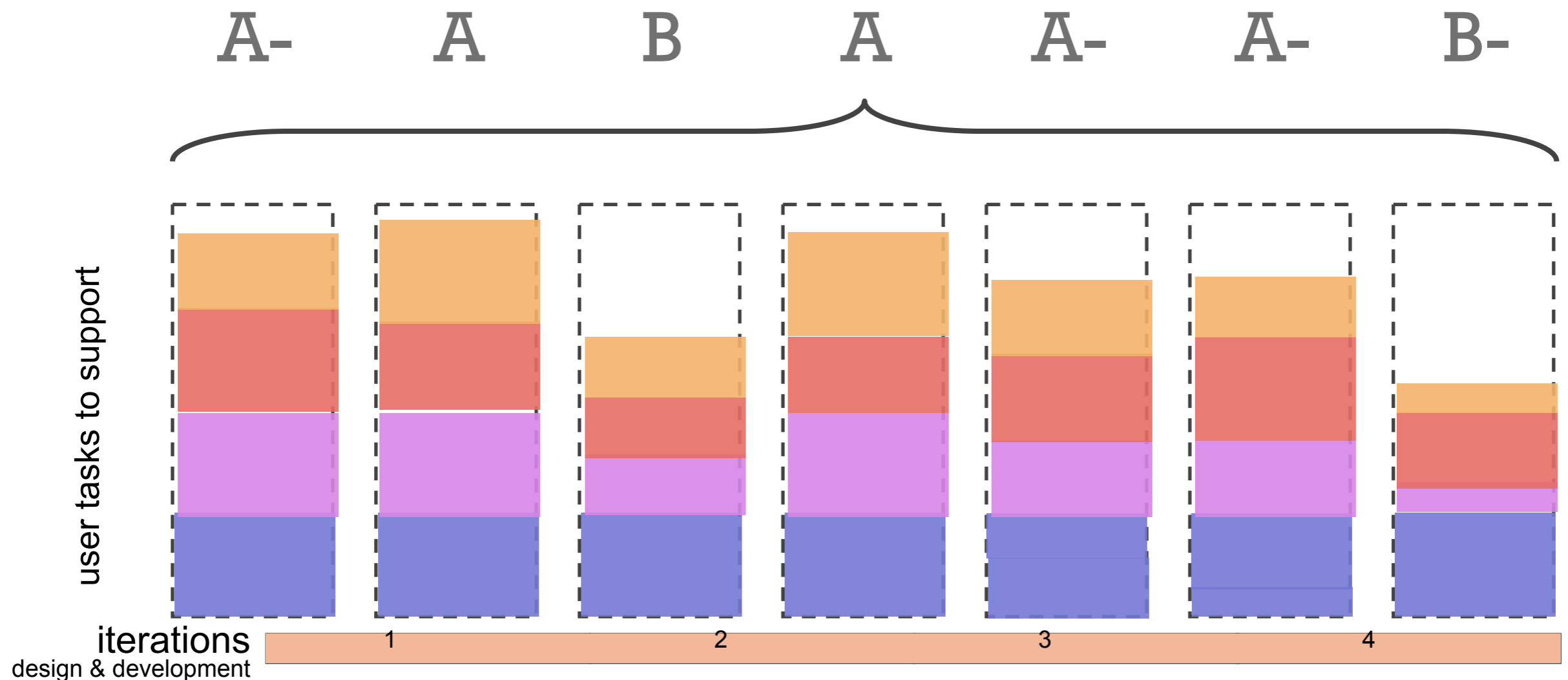




Pete and Roger know that each bus feature can be split into user stories based on quality characteristics.

In early iterations Pete and Roger focus on necessity, then and flexibility and safety, then finish off with luxury

At each iteration they give their features a quality grade, then evaluate their bus report card.





These strategies make sense, but implementing them is hard. Are there any tools that can help me?



This is **Jeff**





This is **Jeff**

He created several of the slides in this presentation.



## This is **Jeff**

He created several of the slides in this presentation.

**(He's a smart guy.)**

Inspired by Constantine & Lockwood's Task Modeling and story writing workshops, Jeff came up with a tool that helps us implement the three strategies.

The tool is called **User Story Mapping**.



Understand who is involved and why

# Understand who is involved and why



My goal is to keep my teeth healthy

# Understand who is involved and why

Patient

# Understand who is involved and why

Patient

My goal is to help people keep their teeth healthy + run an efficient office



# Understand who is involved and why

Patient

Dentist

# Understand who is involved and why

Patient

Dentist

My goal is to help the dentist help people keep their teeth healthy



# Understand who is involved and why

Patient

Dental  
assistant

Dentist

# Understand who is involved and why

Patient

Dental  
assistant

Dentist



# Understand what people do

Patient

Dental  
assistant

Dentist

People perform tasks.

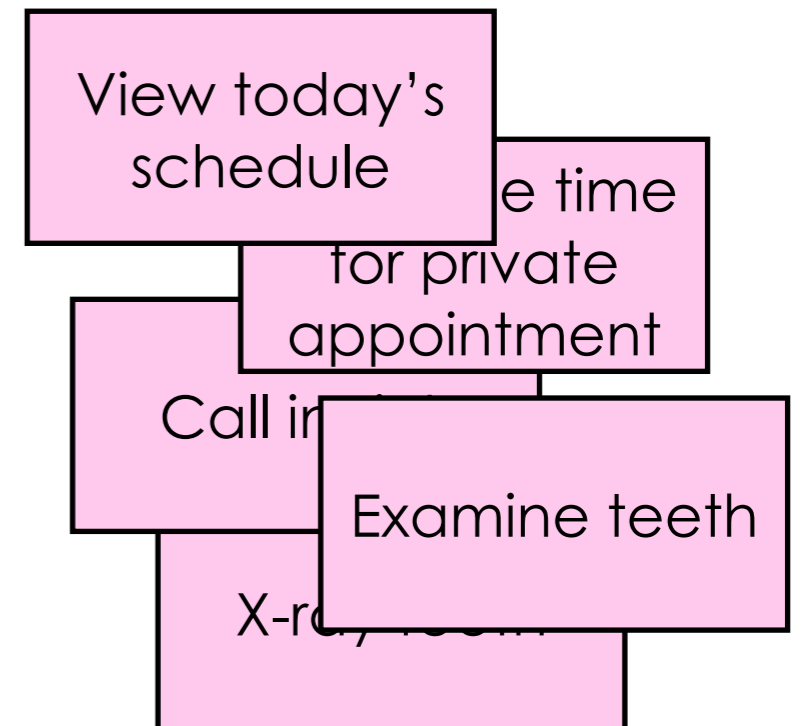
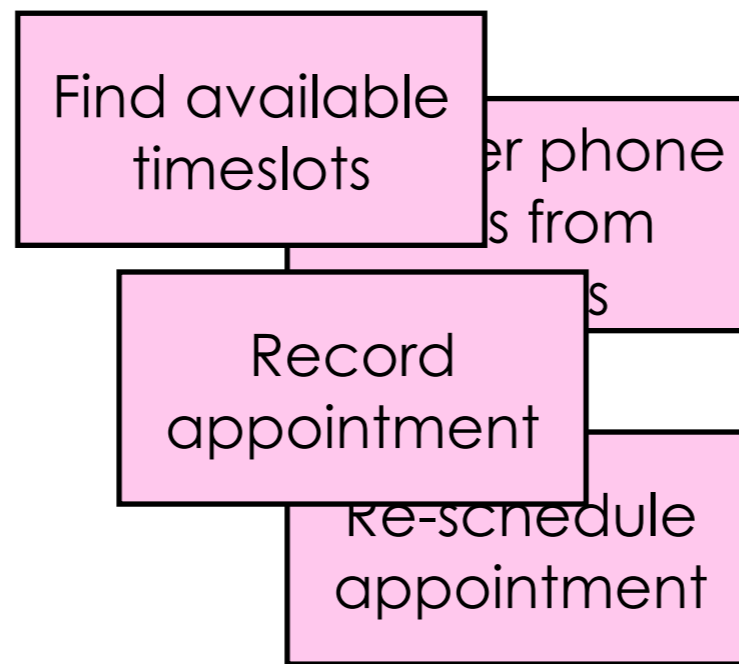
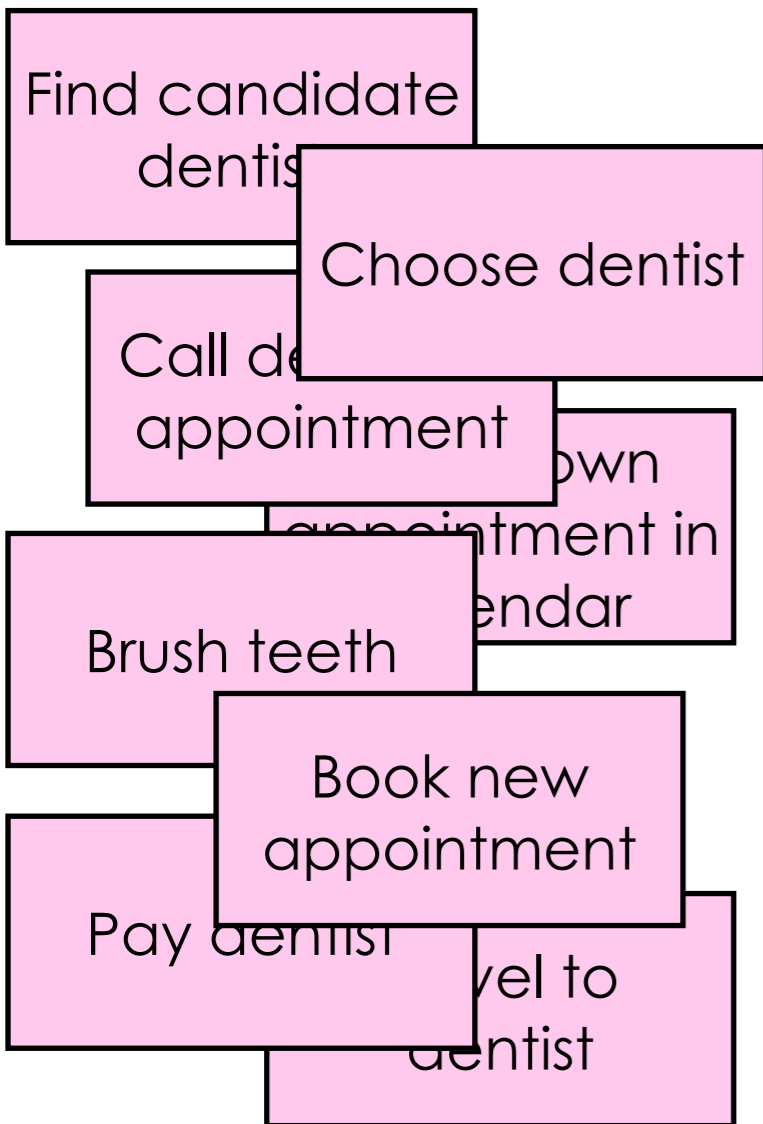
Tasks have an objective  
that can be completed.

# Understand what people do

Patient

Dental  
assistant

Dentist



People perform tasks.

Tasks have an objective that can be completed.

# Understand what people do

Patient

Dental  
assistant

Dentist

Find candidate  
dentists

Choose dentist

Call dentist for  
appointment

Write down  
appointment in  
calendar

Brush teeth

Book new  
appointment

Pay dentist

Travel to  
dentist

Find available  
timeslots

Answer phone  
calls from  
clients

Record  
appointment

Re-schedule  
appointment

View today's  
schedule

Schedule time  
for private  
appointment

Call in sick

Do stuff

Do more stuff

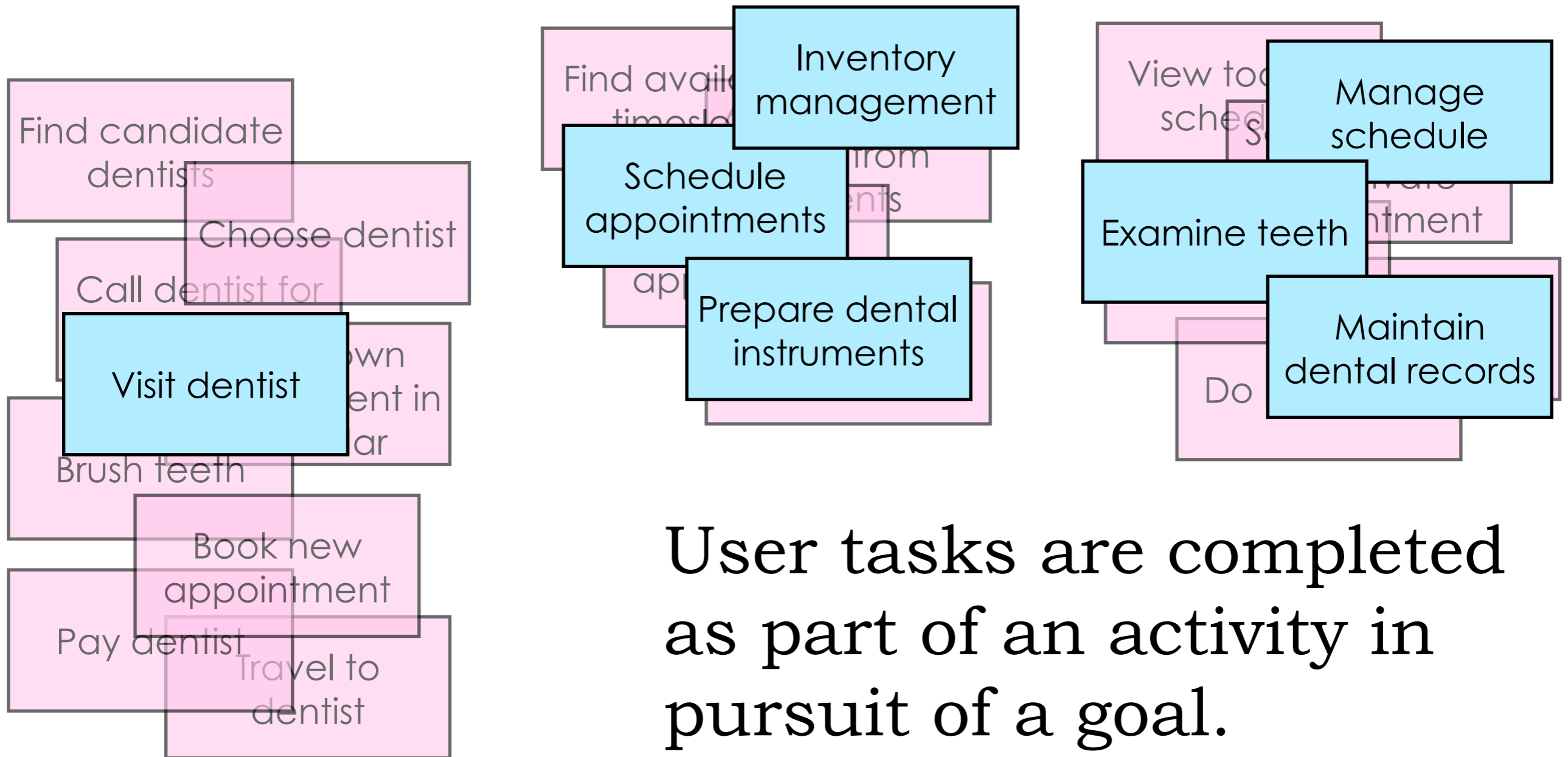
**User tasks are completed  
as part of an activity in  
pursuit of a goal.**

# Understand what people do

Patient

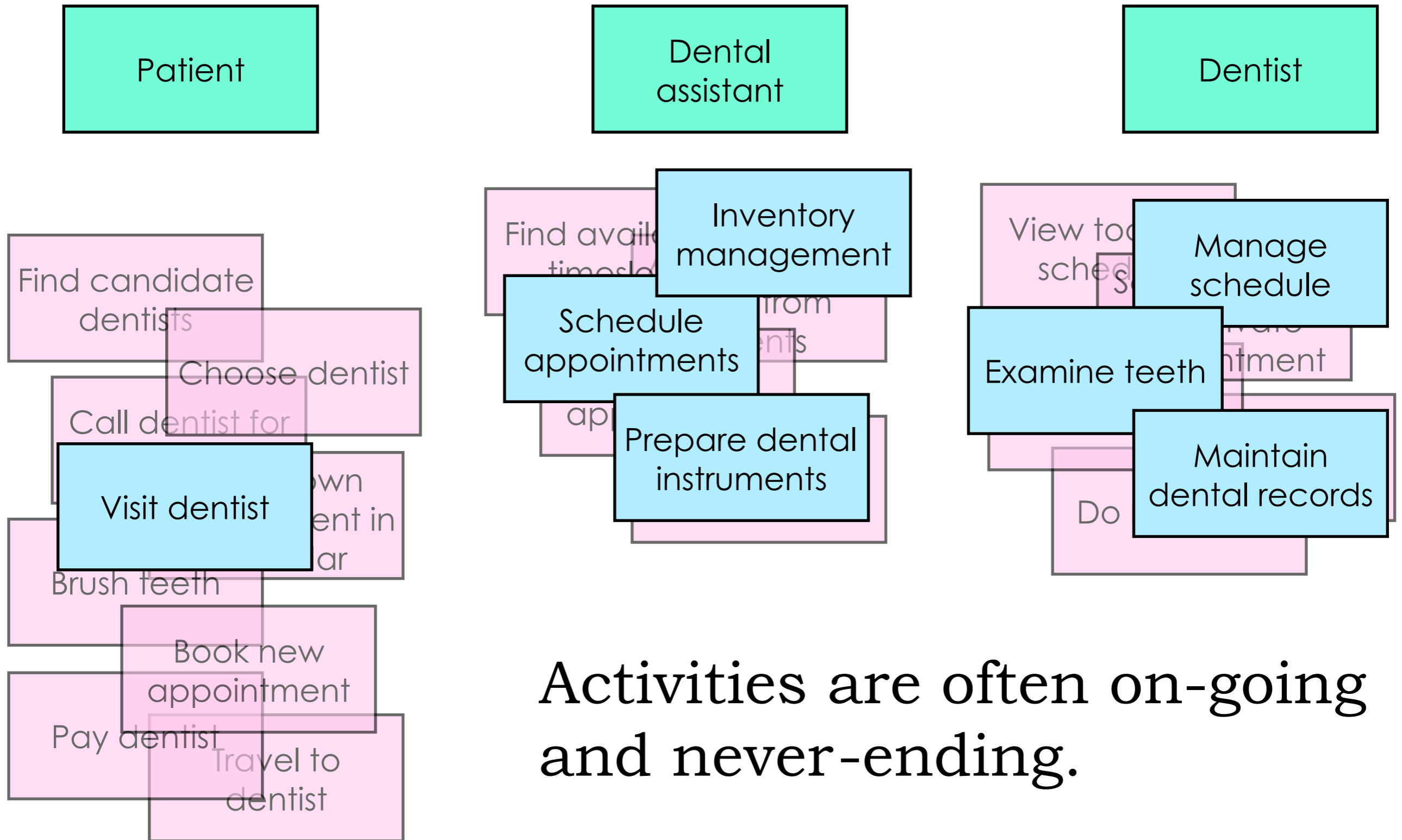
Dental  
assistant

Dentist



User tasks are completed as part of an activity in pursuit of a goal.

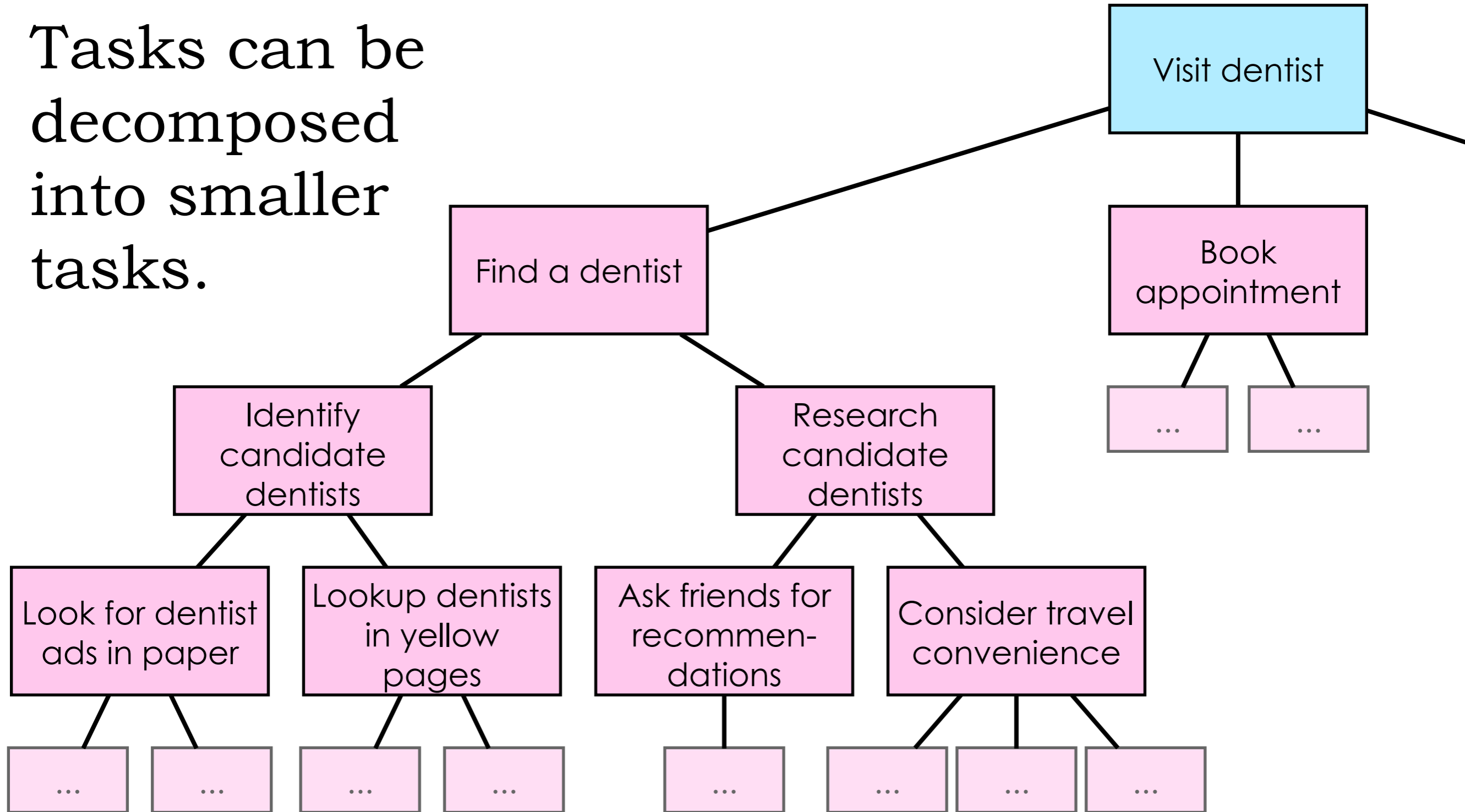
# Understand what people do



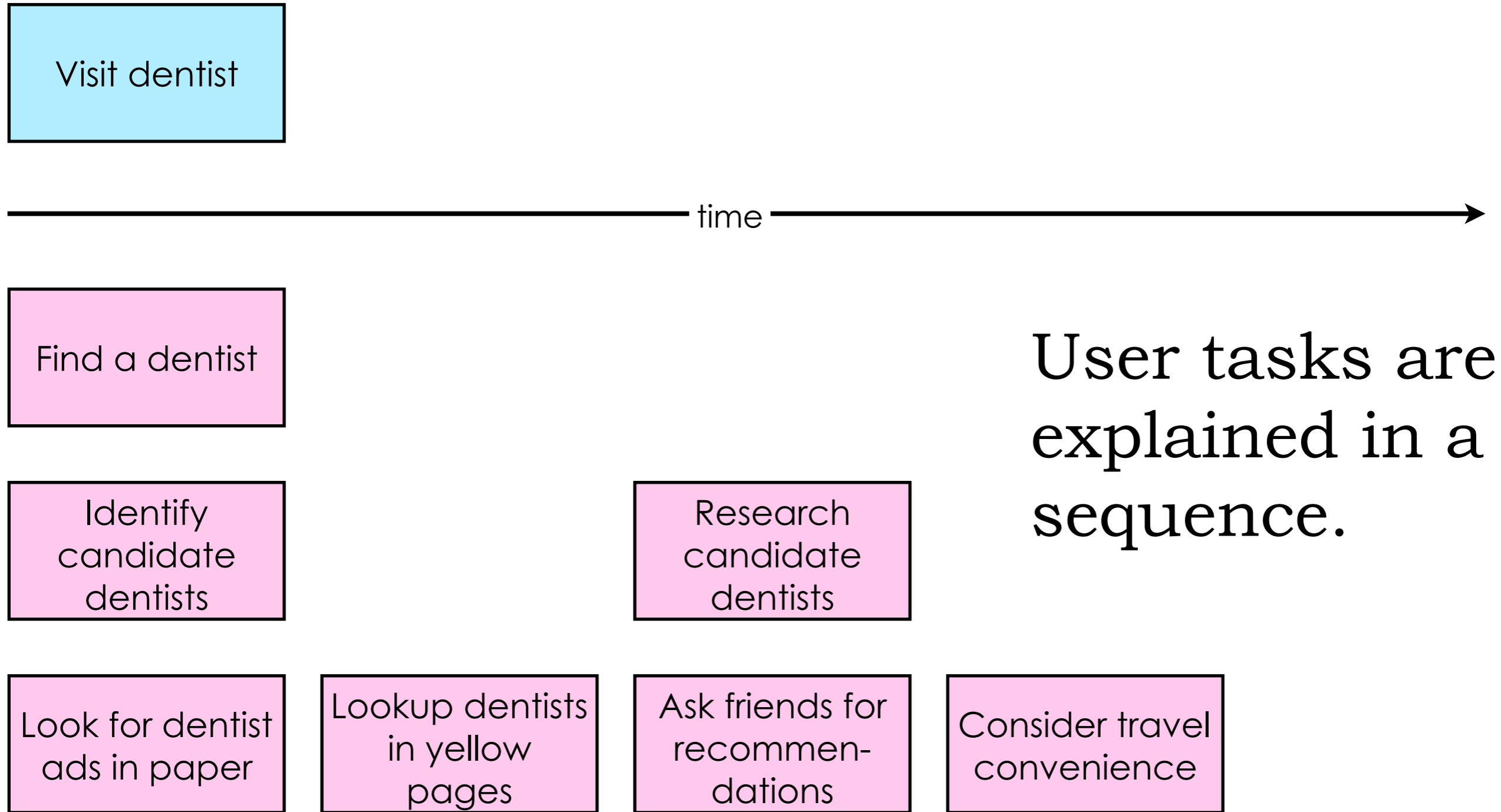
**Activities are often on-going and never-ending.**

# Understand what people do

Tasks can be decomposed into smaller tasks.



# Understand what people do



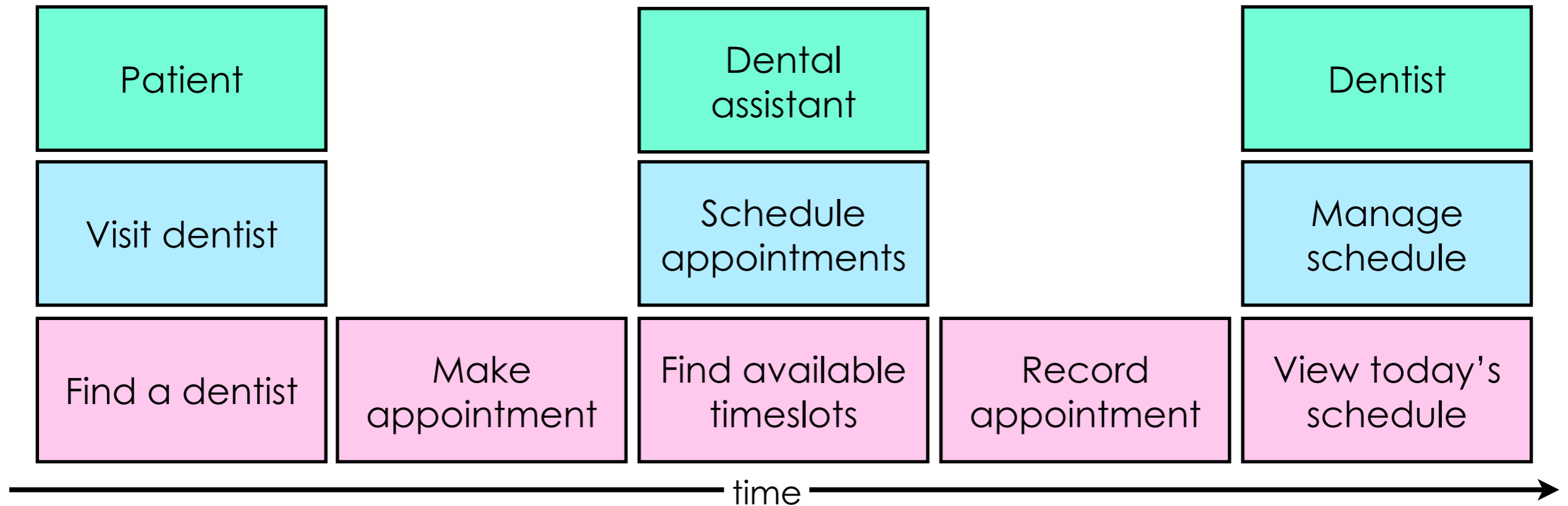
# Understand what people do



People's activities and tasks  
interact to create workflows.

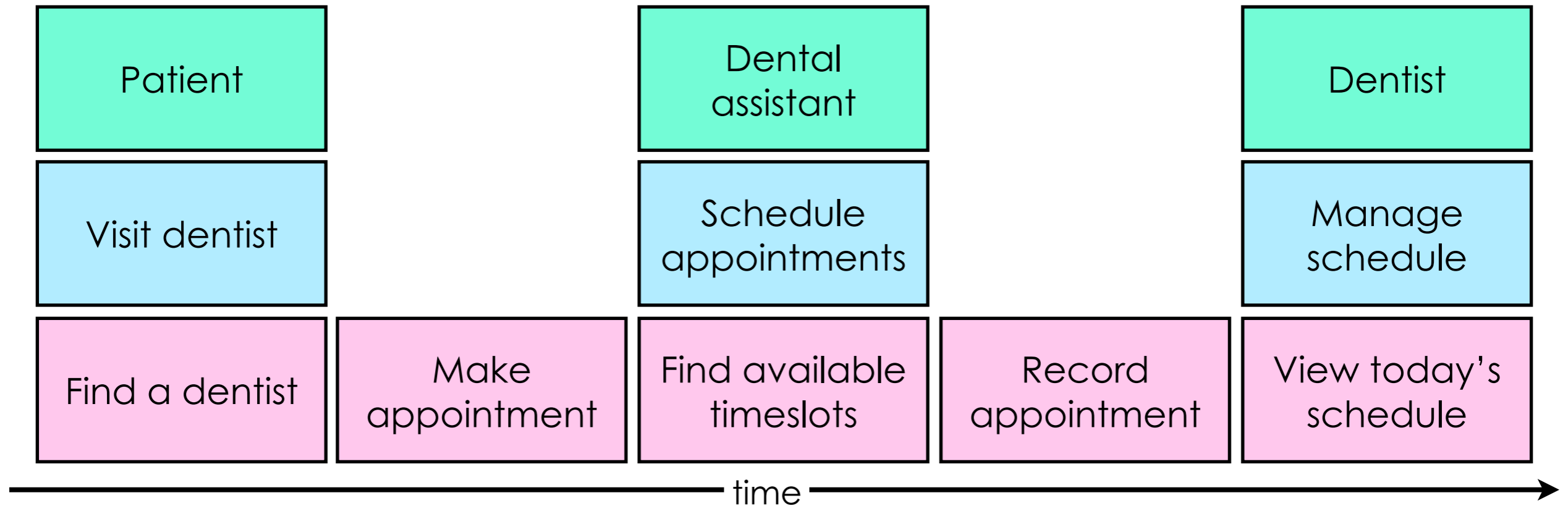


# Understand what people do



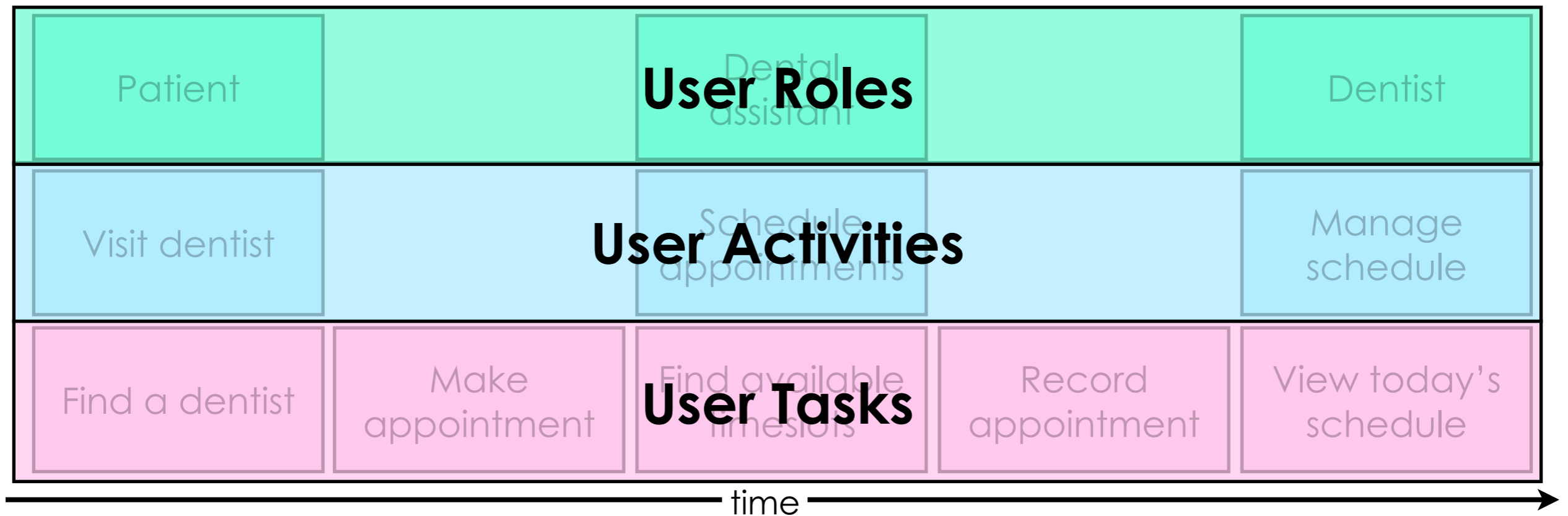
People's activities and tasks interact to create workflows.

# Understand what people do



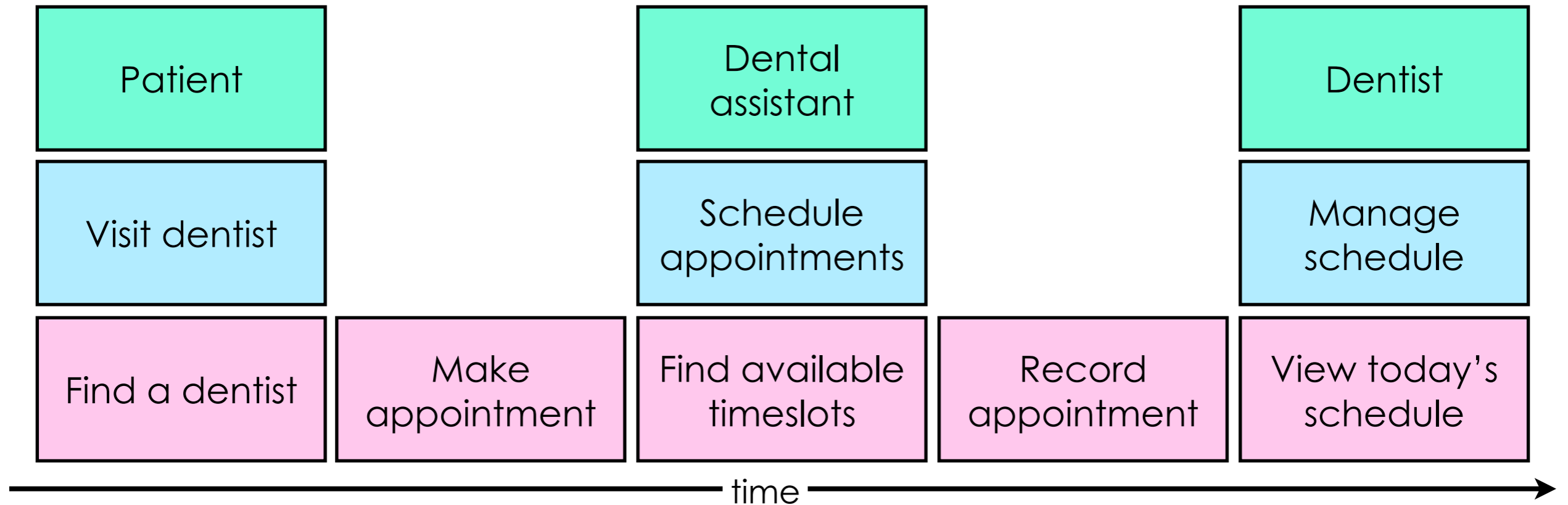
Roles, Activities and Task cards can be organised in a User Story Map.

# Understand what people do



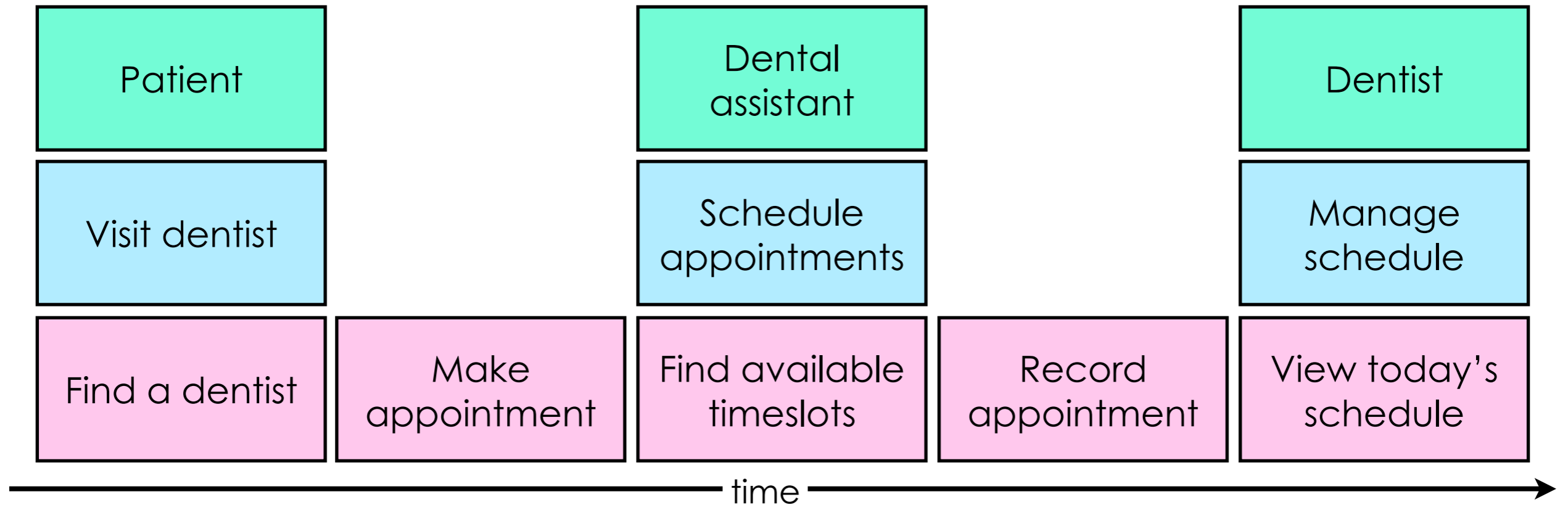
Roles, Activities and Task cards can be organised in a User Story Map.

# Story maps help tell the stories



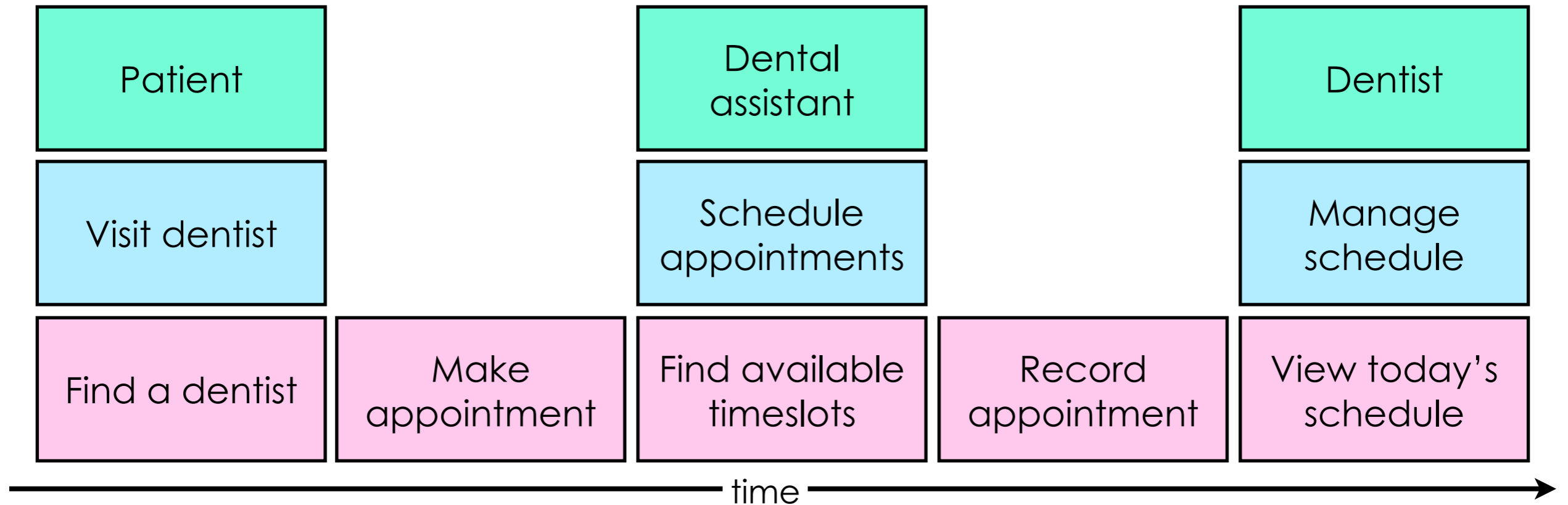
Both short and long version.

# Story maps help tell the stories



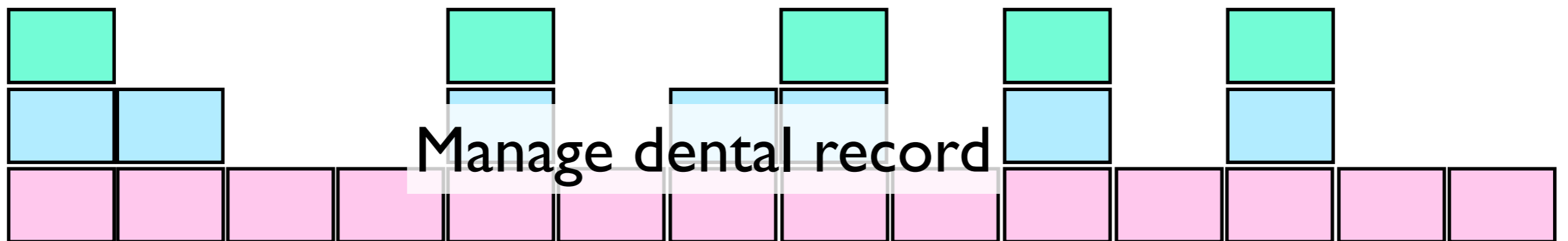
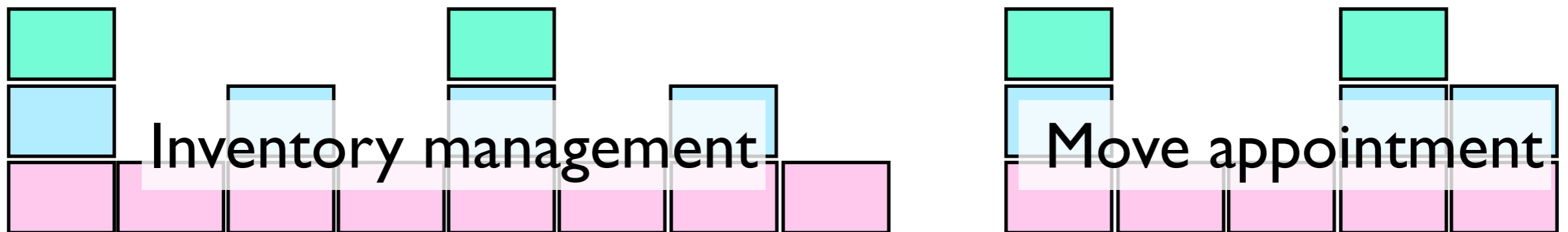
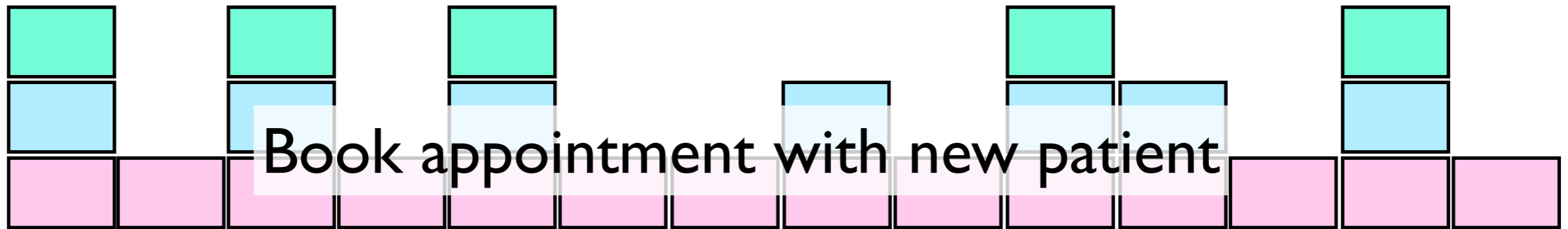
Both short and long version.

# Story maps help tell the stories

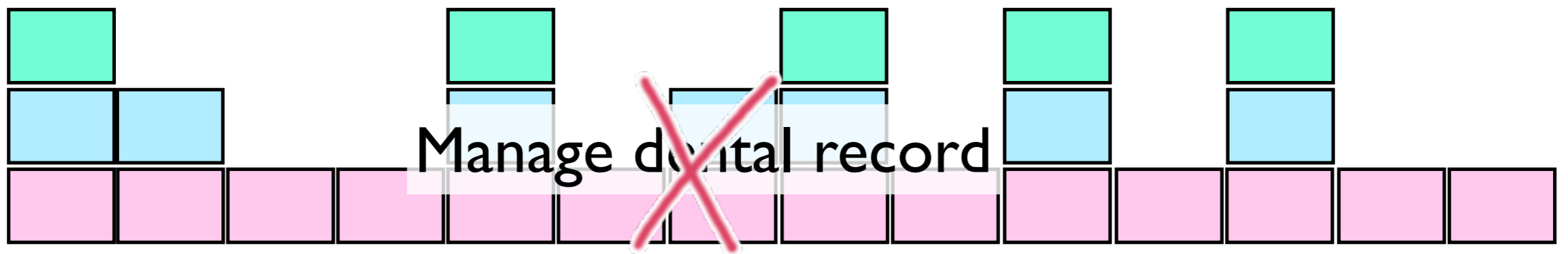
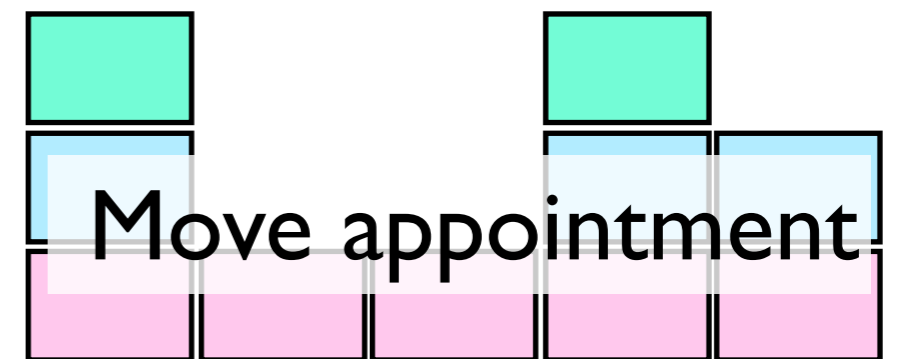
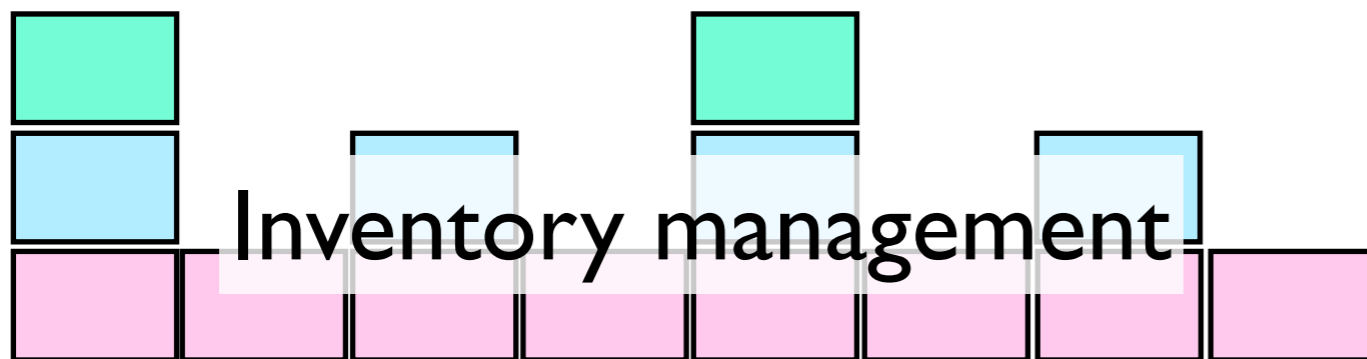
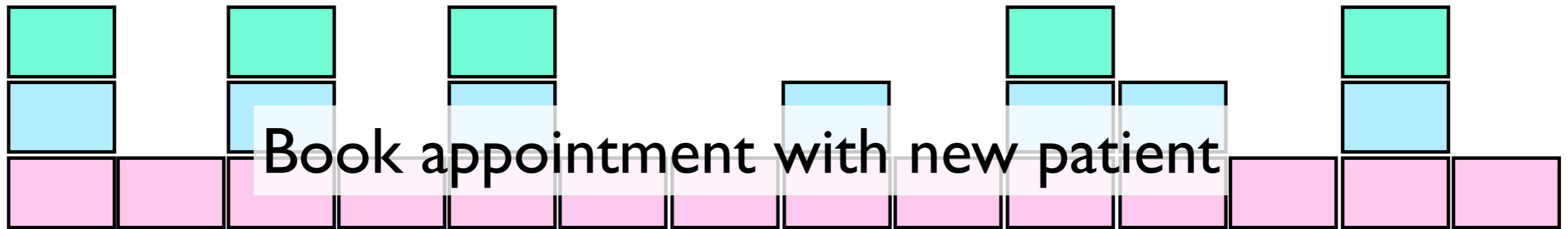


Both short and long version.

# Prioritize workflows to support

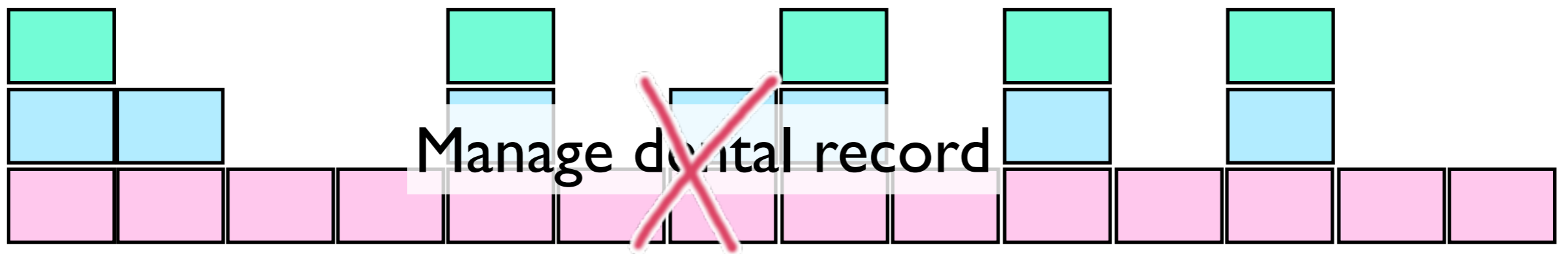
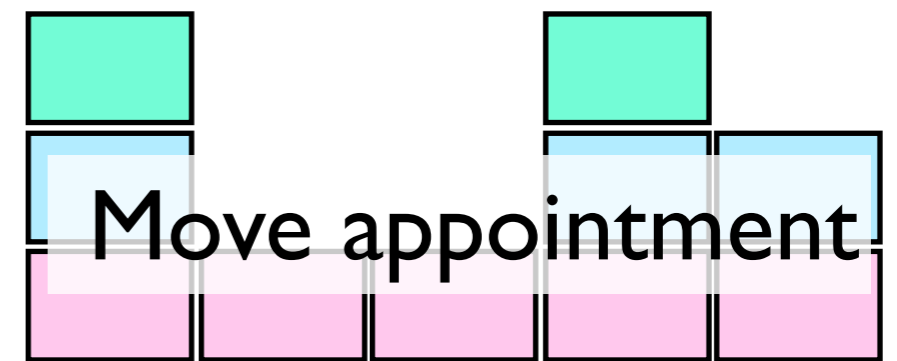
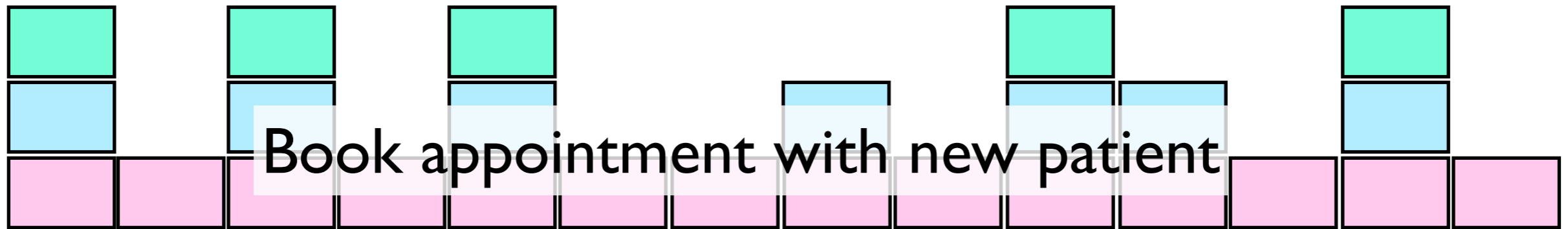


# Prioritize workflows to support

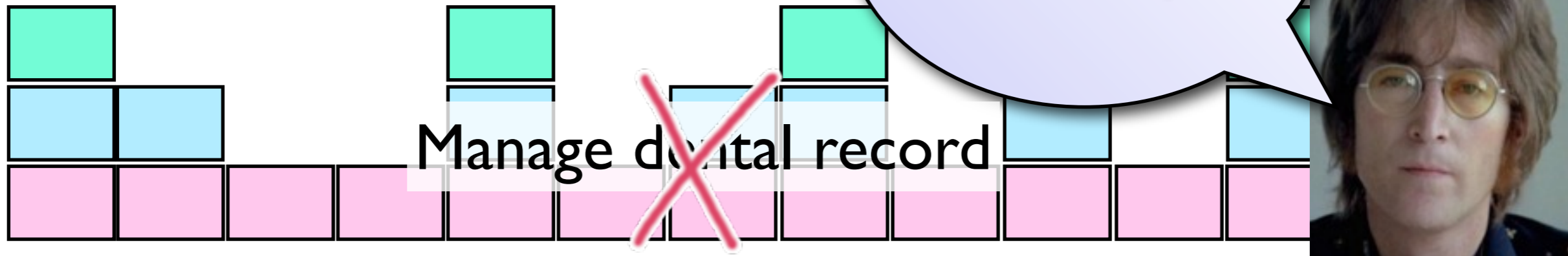
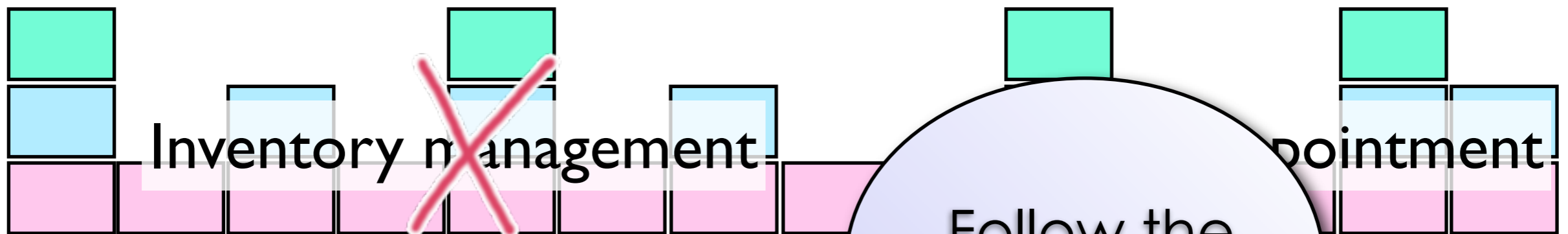
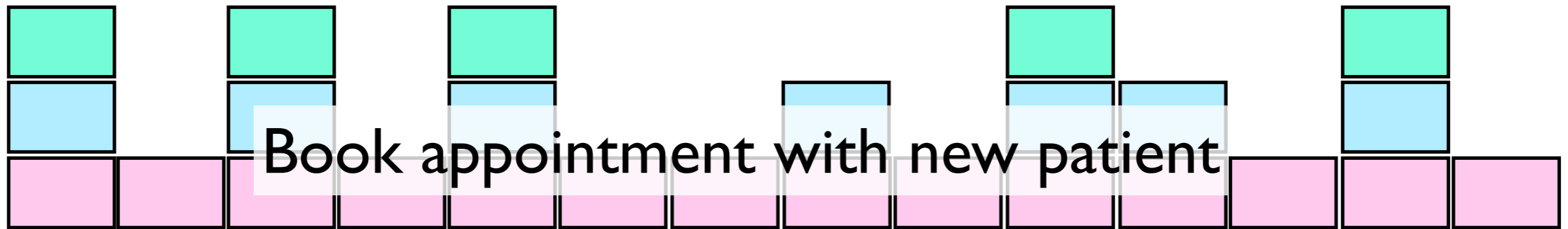




# Prioritize workflows to support



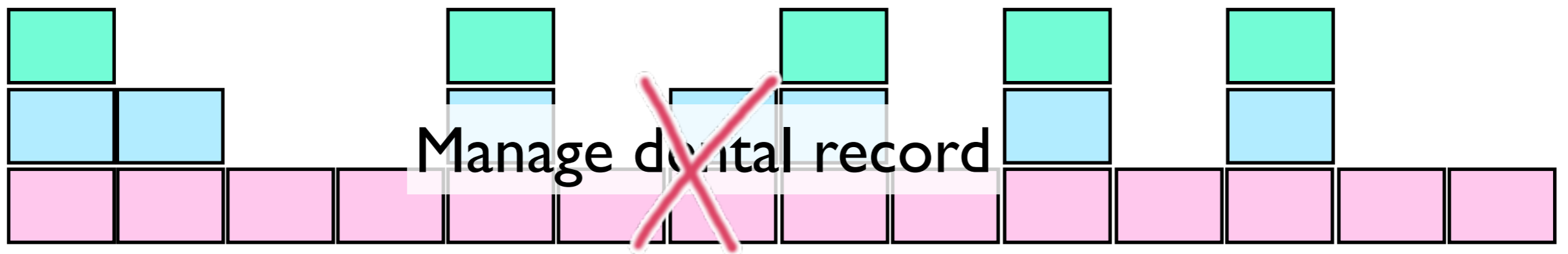
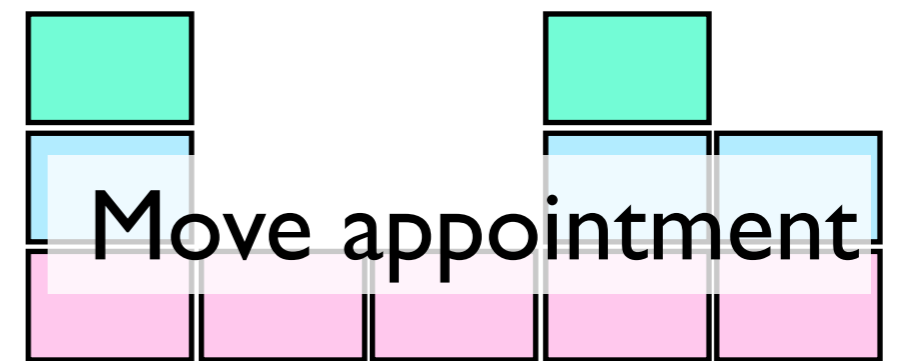
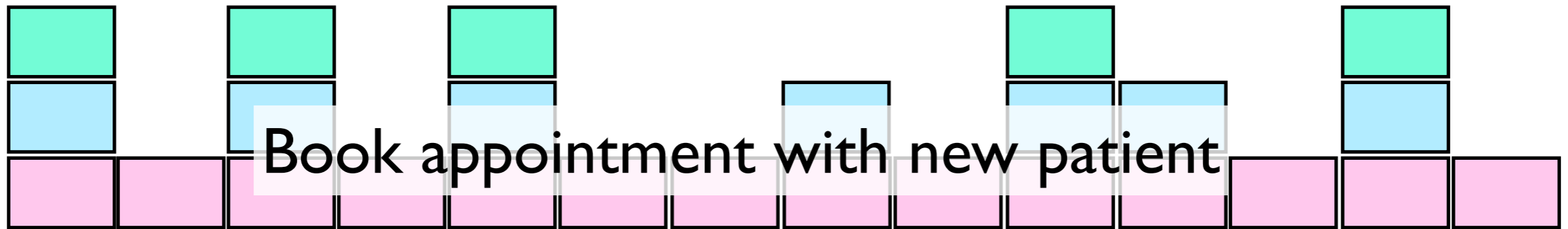
# Prioritize workflows to support



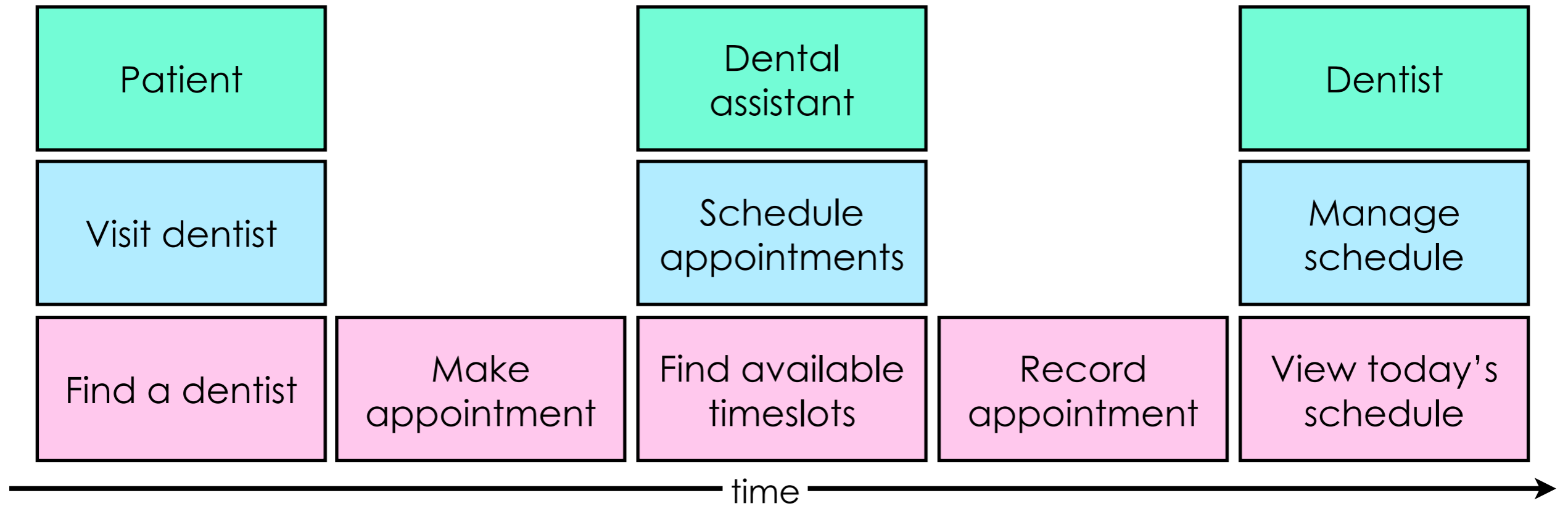
Follow the money



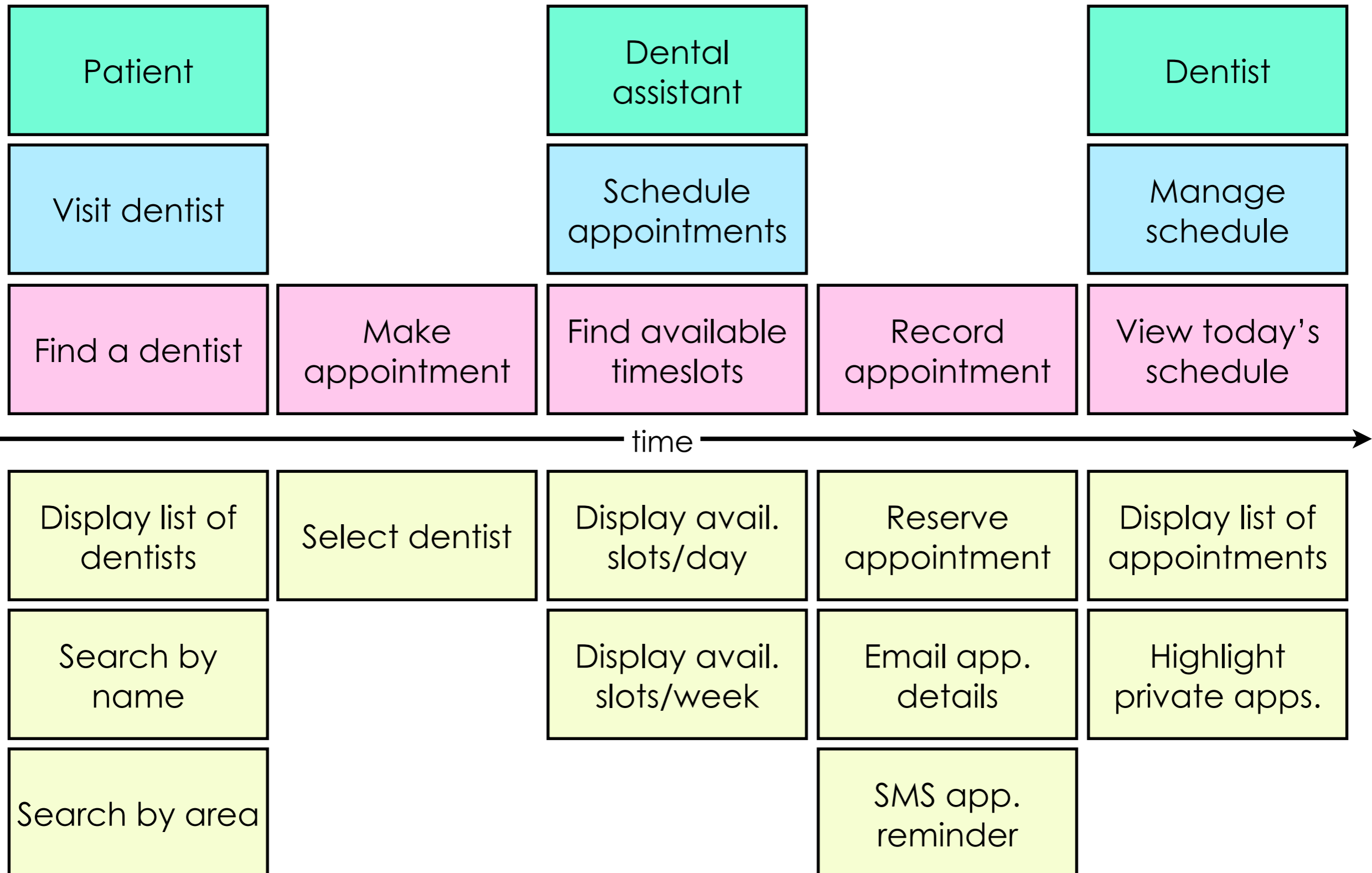
# Prioritize workflows to support



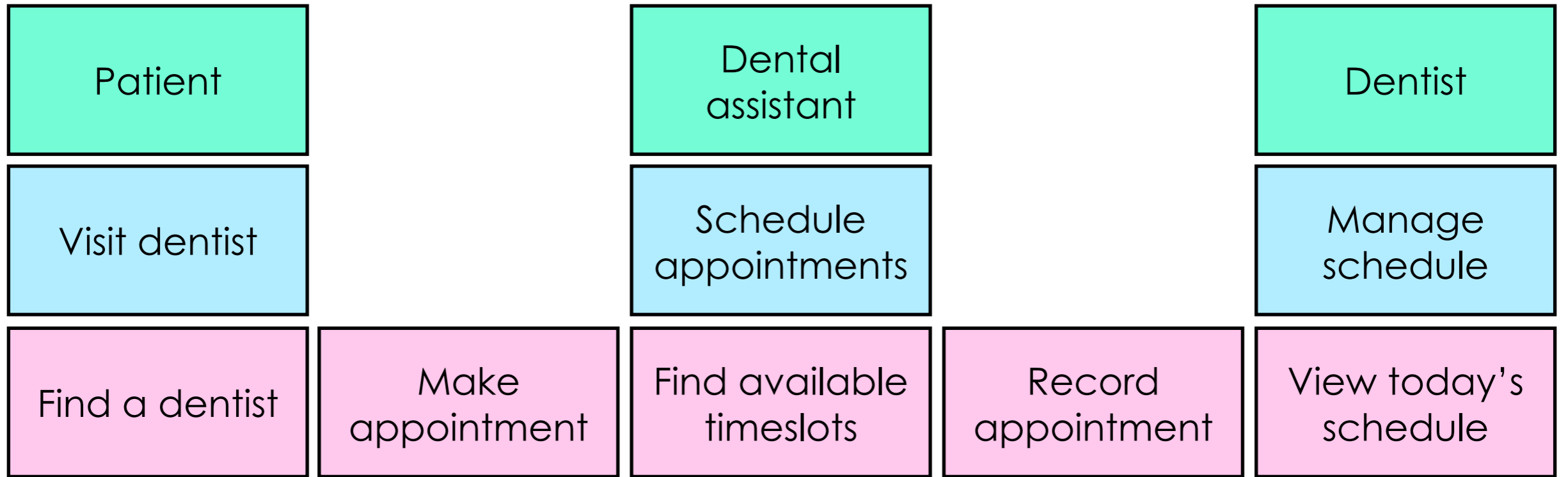
# Identify features



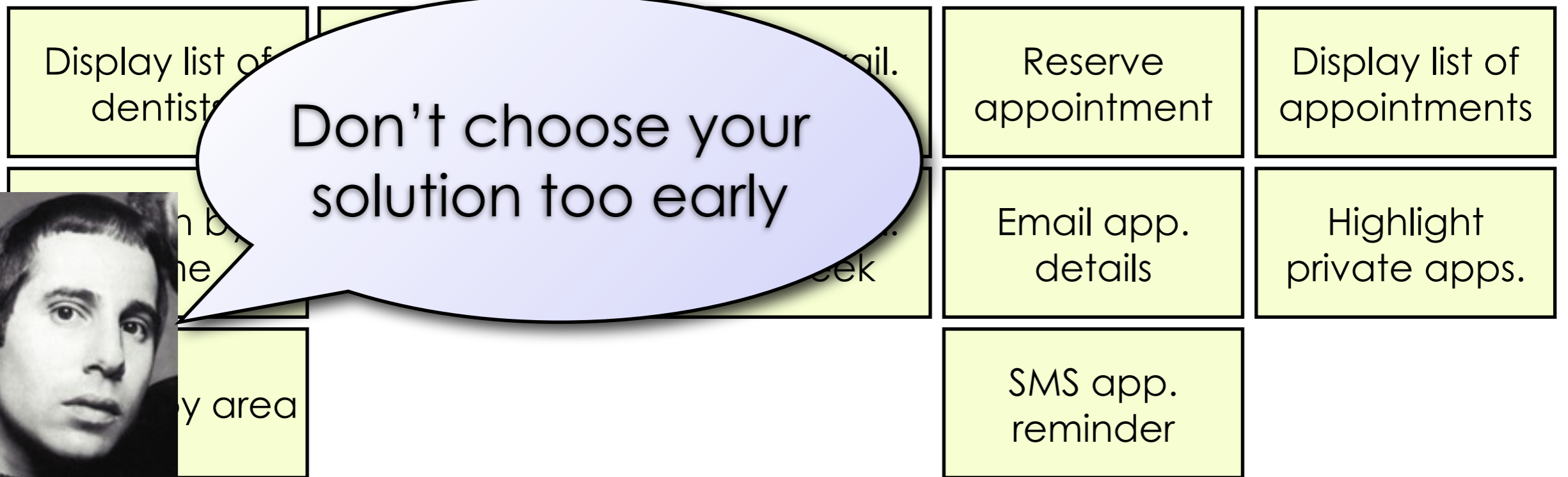
# Identify features



# Identify features



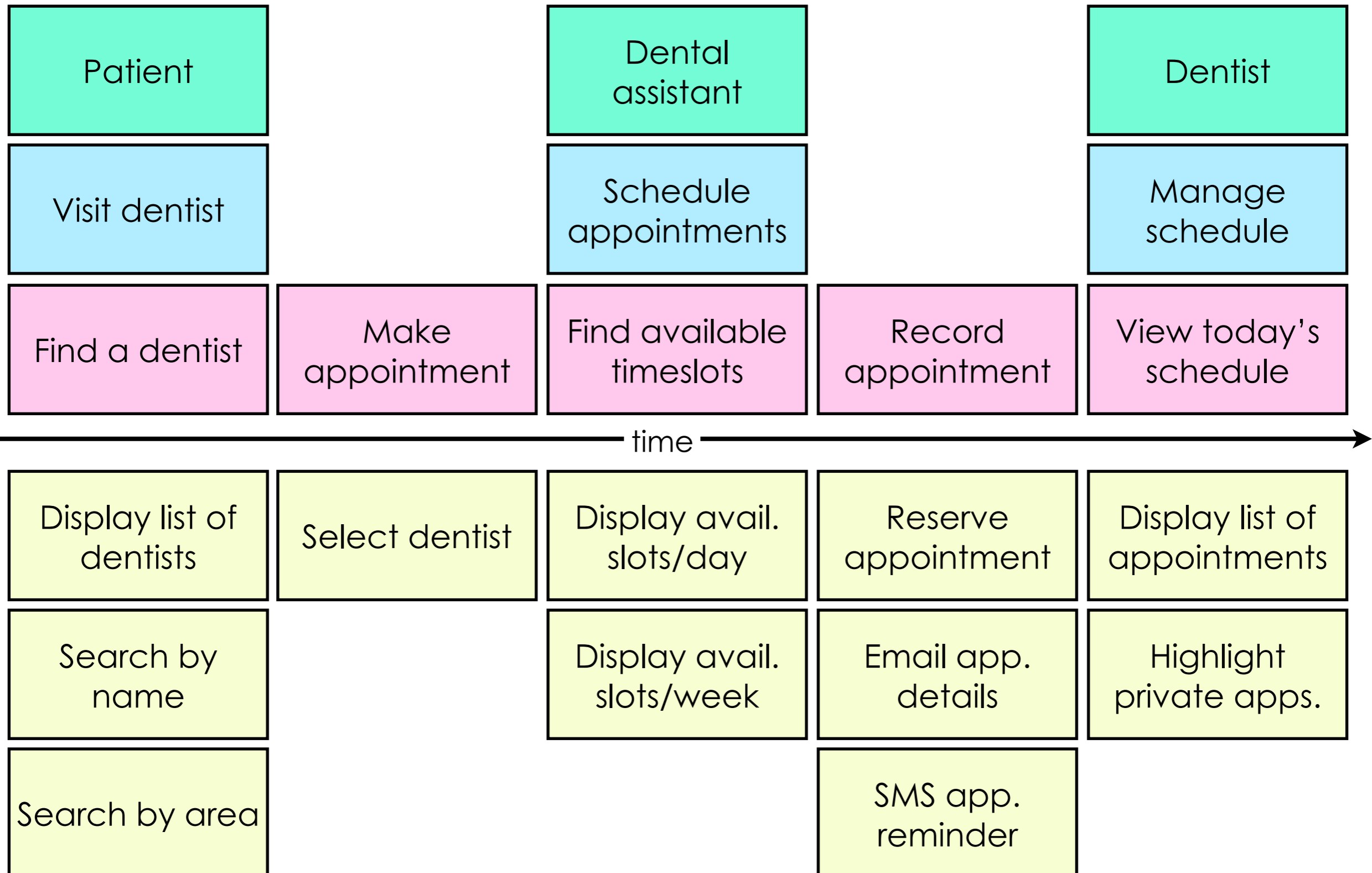
time →



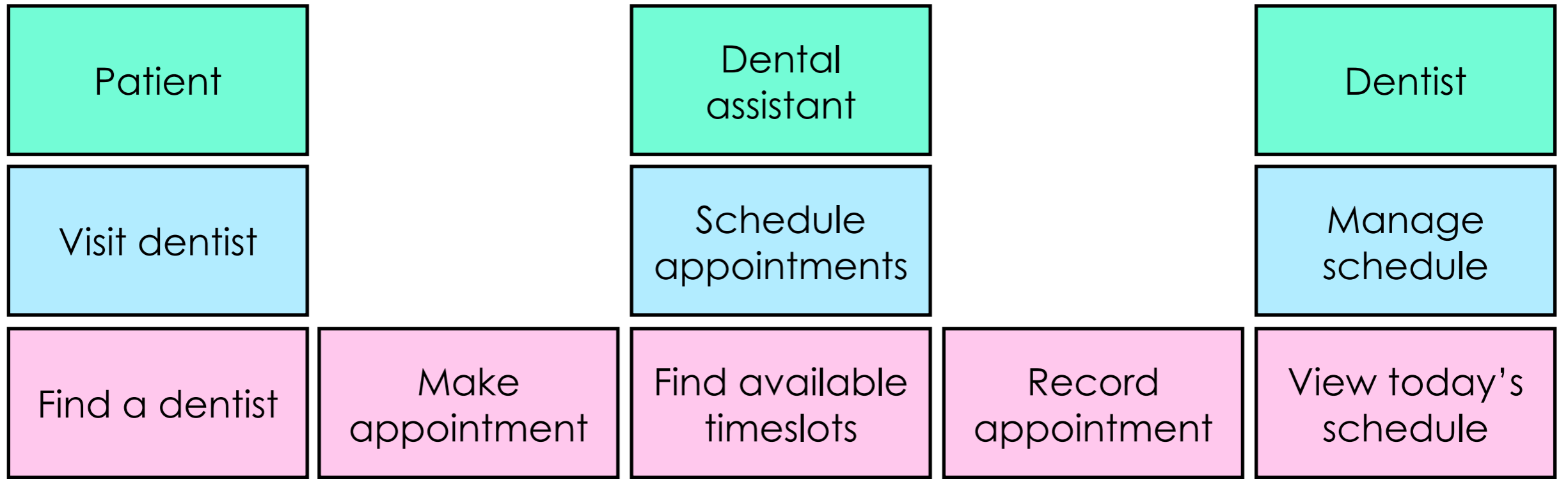
Don't choose your solution too early



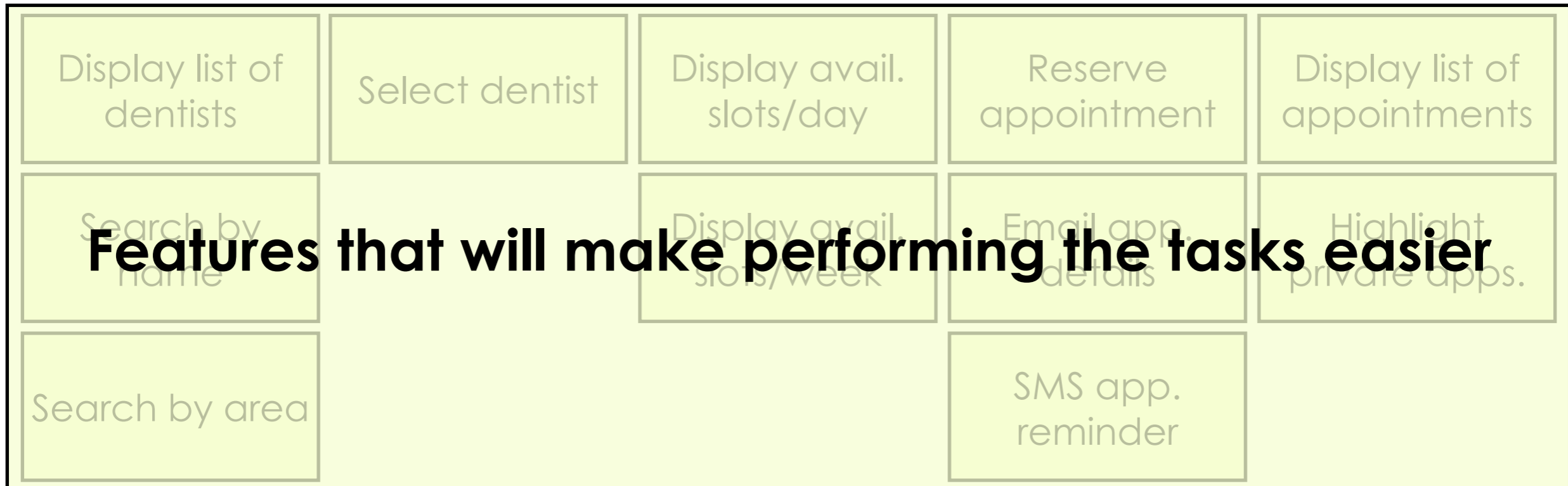
# Identify features



# Identify features



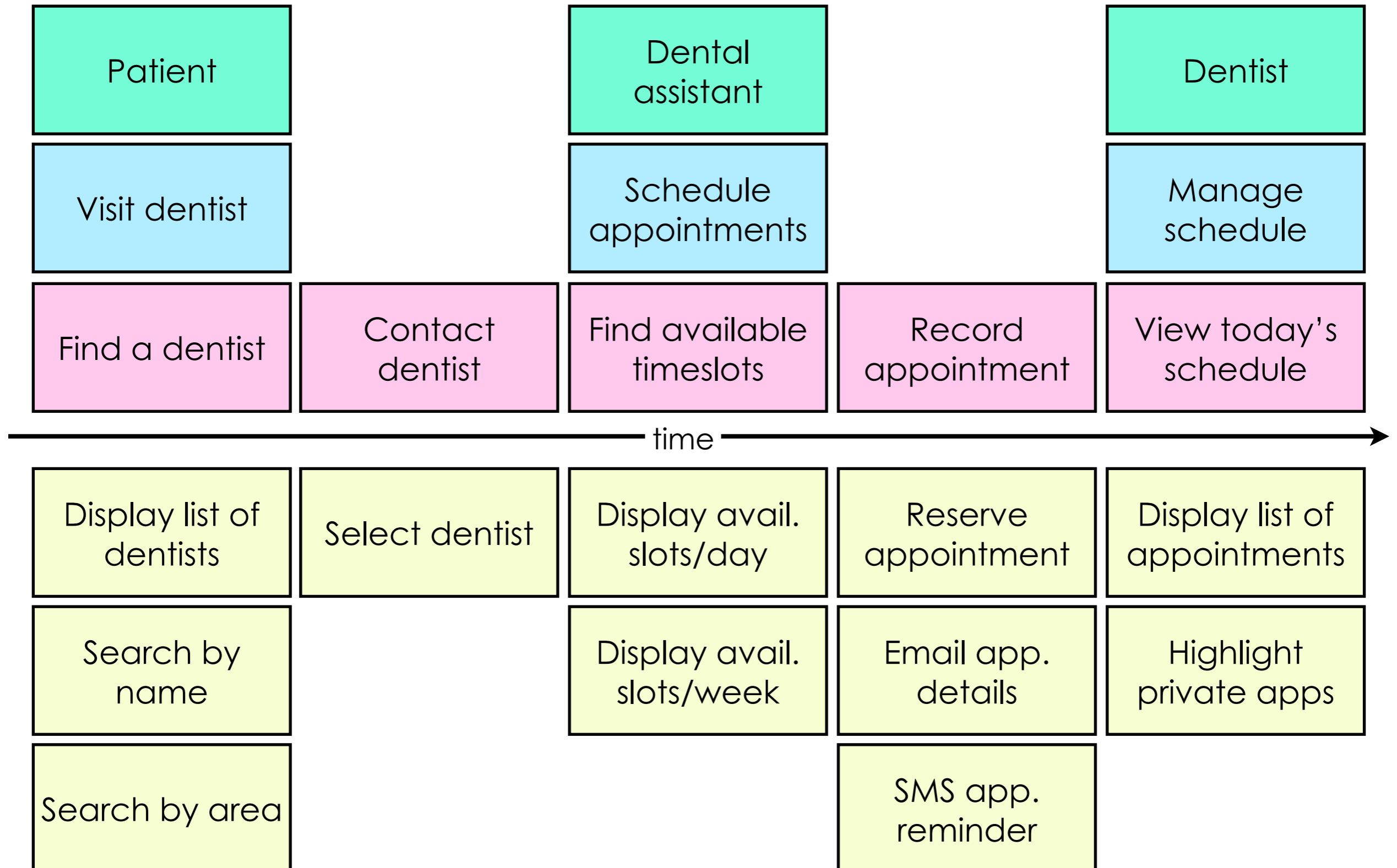
time →



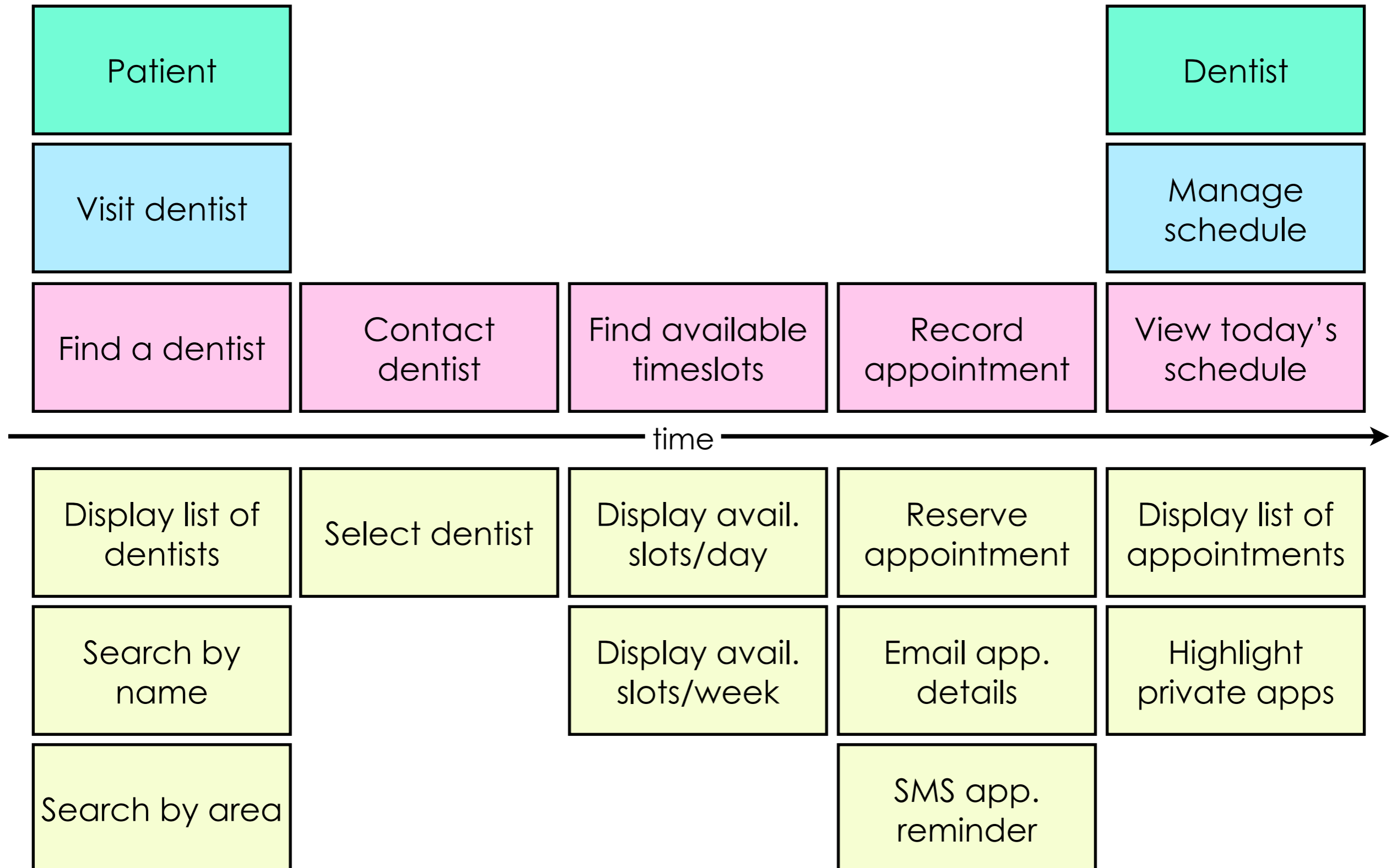
**Features that will make performing the tasks easier**



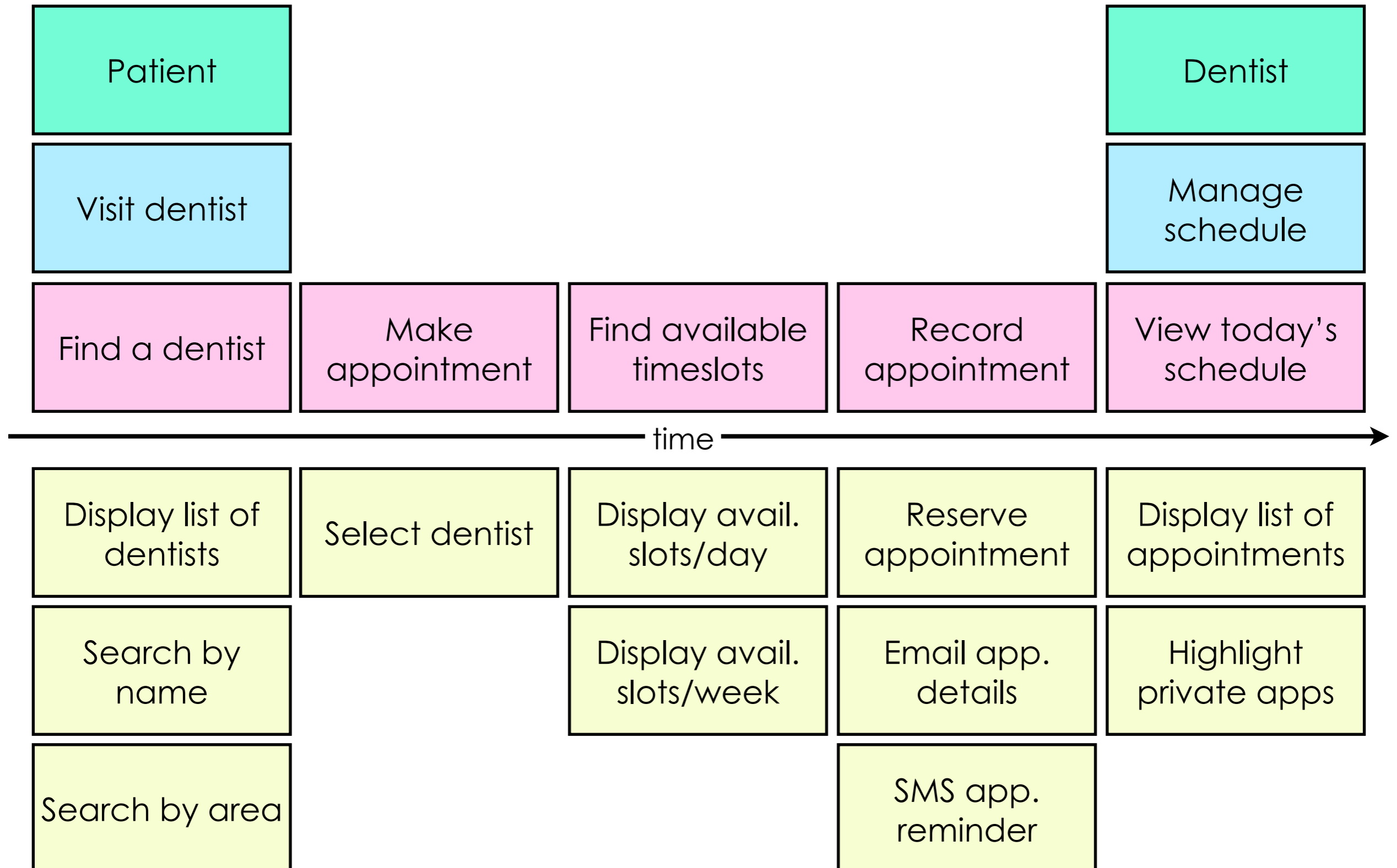
# New tools can change the workflow



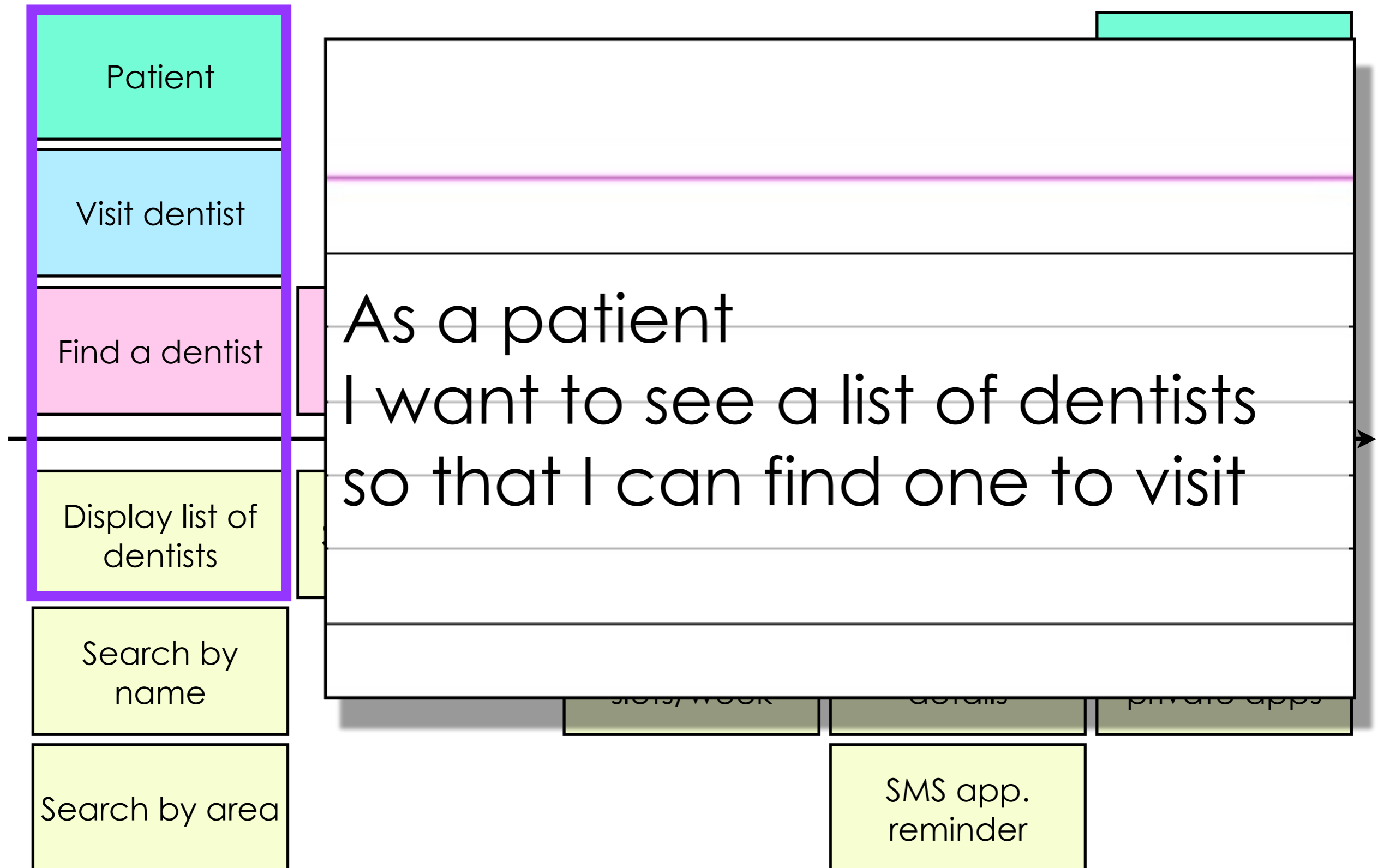
# New tools can change the workflow



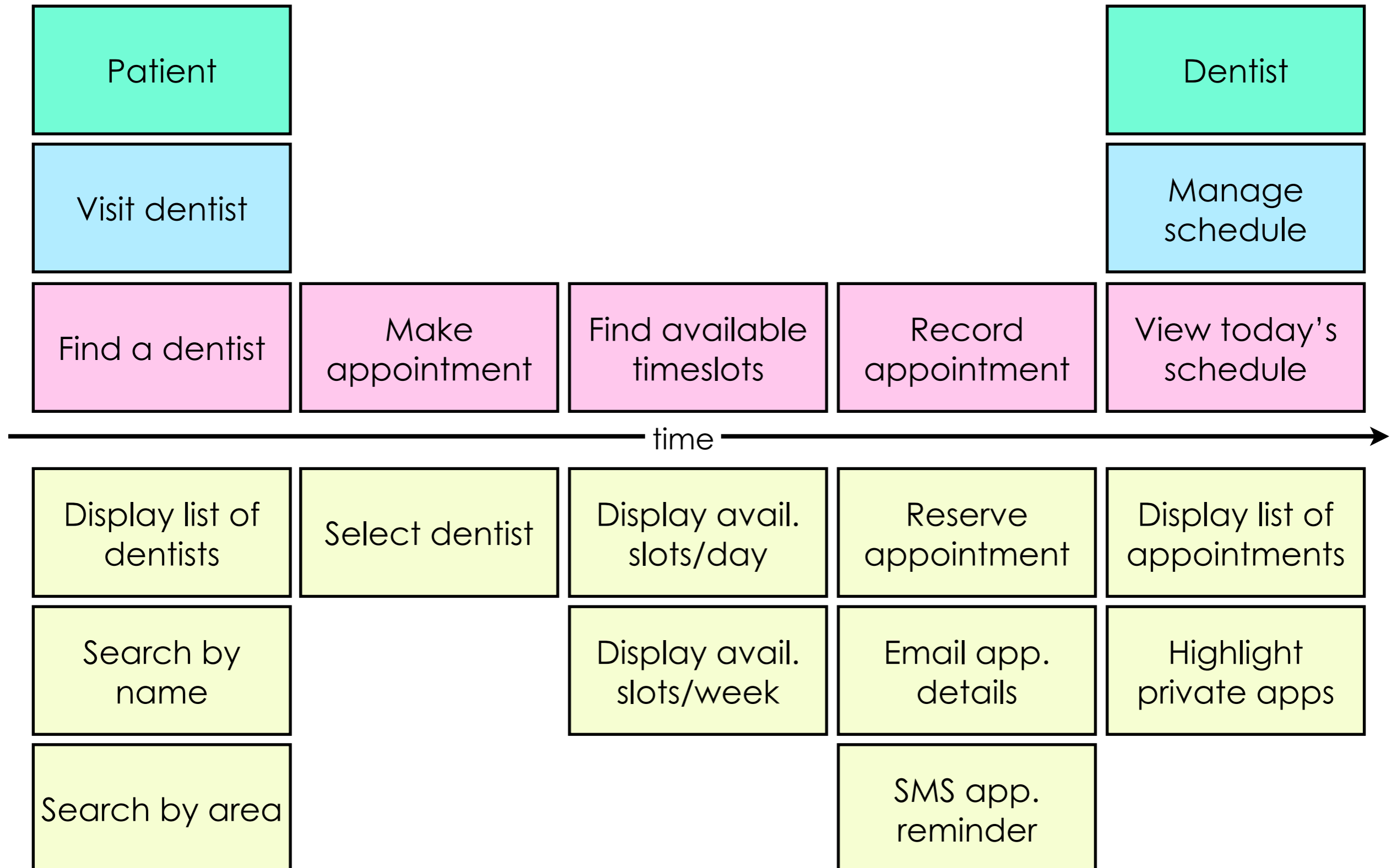
# Story maps provide more context than traditional user stories



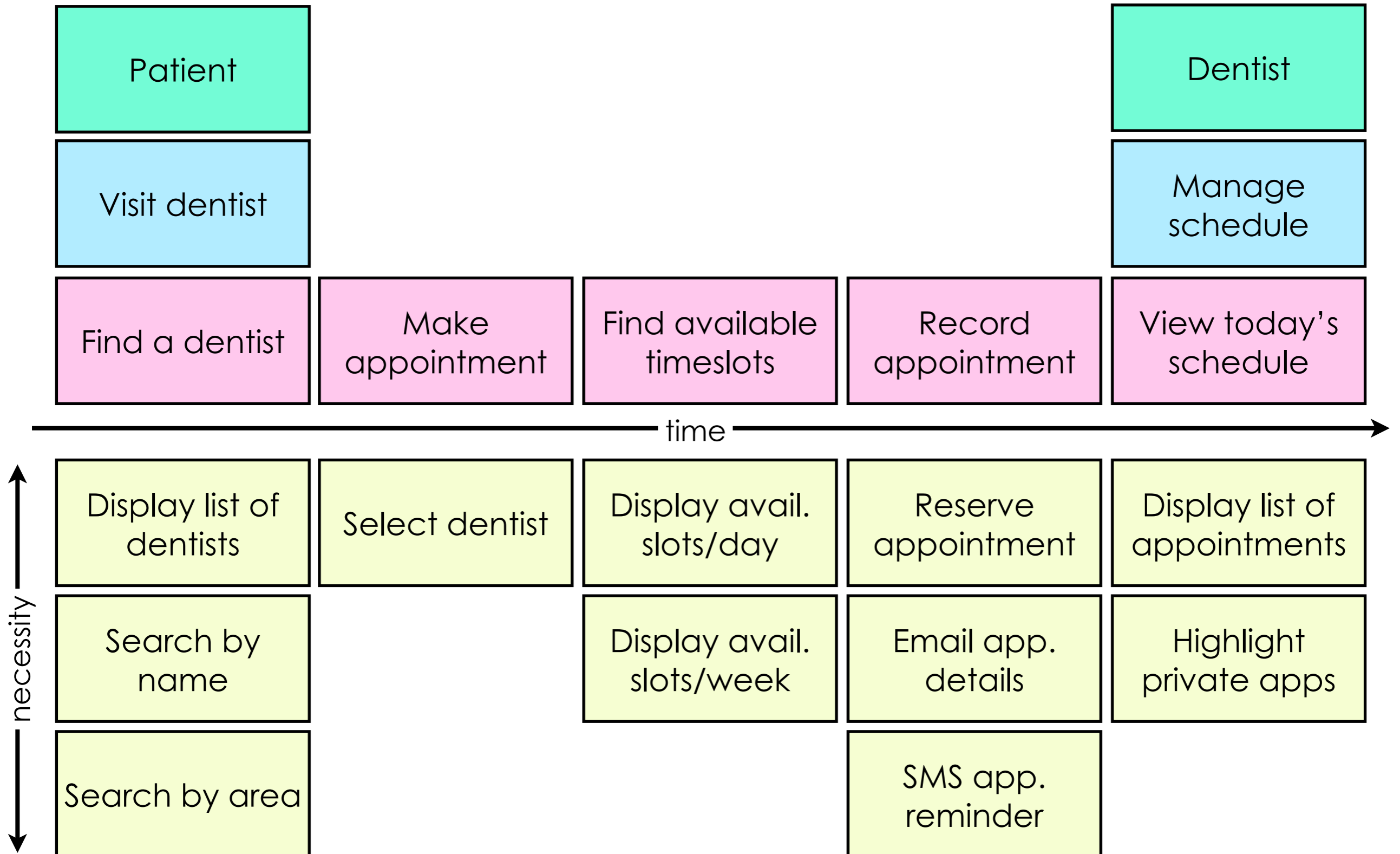
# Story maps provide more context than traditional user stories



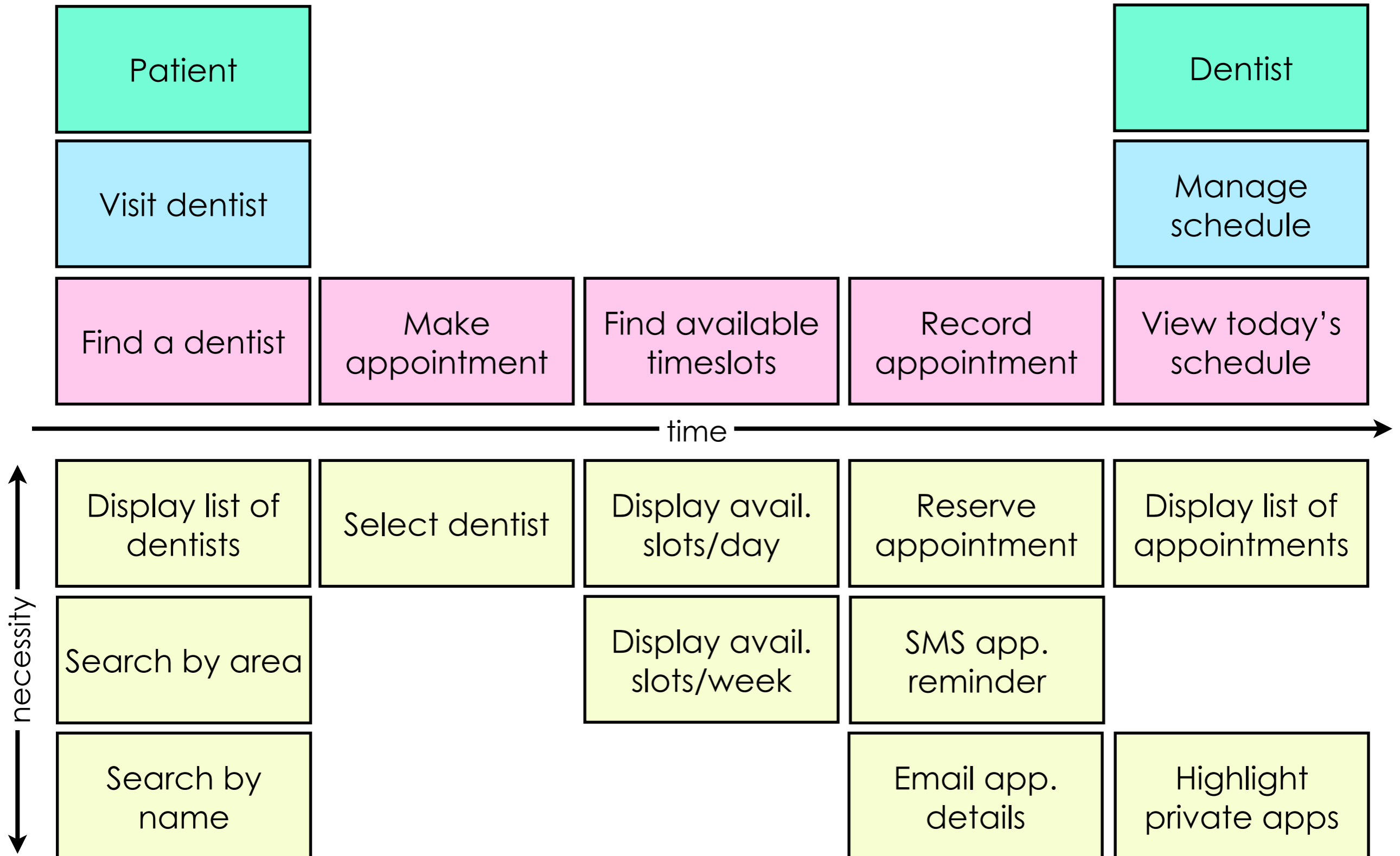
# Story maps provide more context than traditional user stories



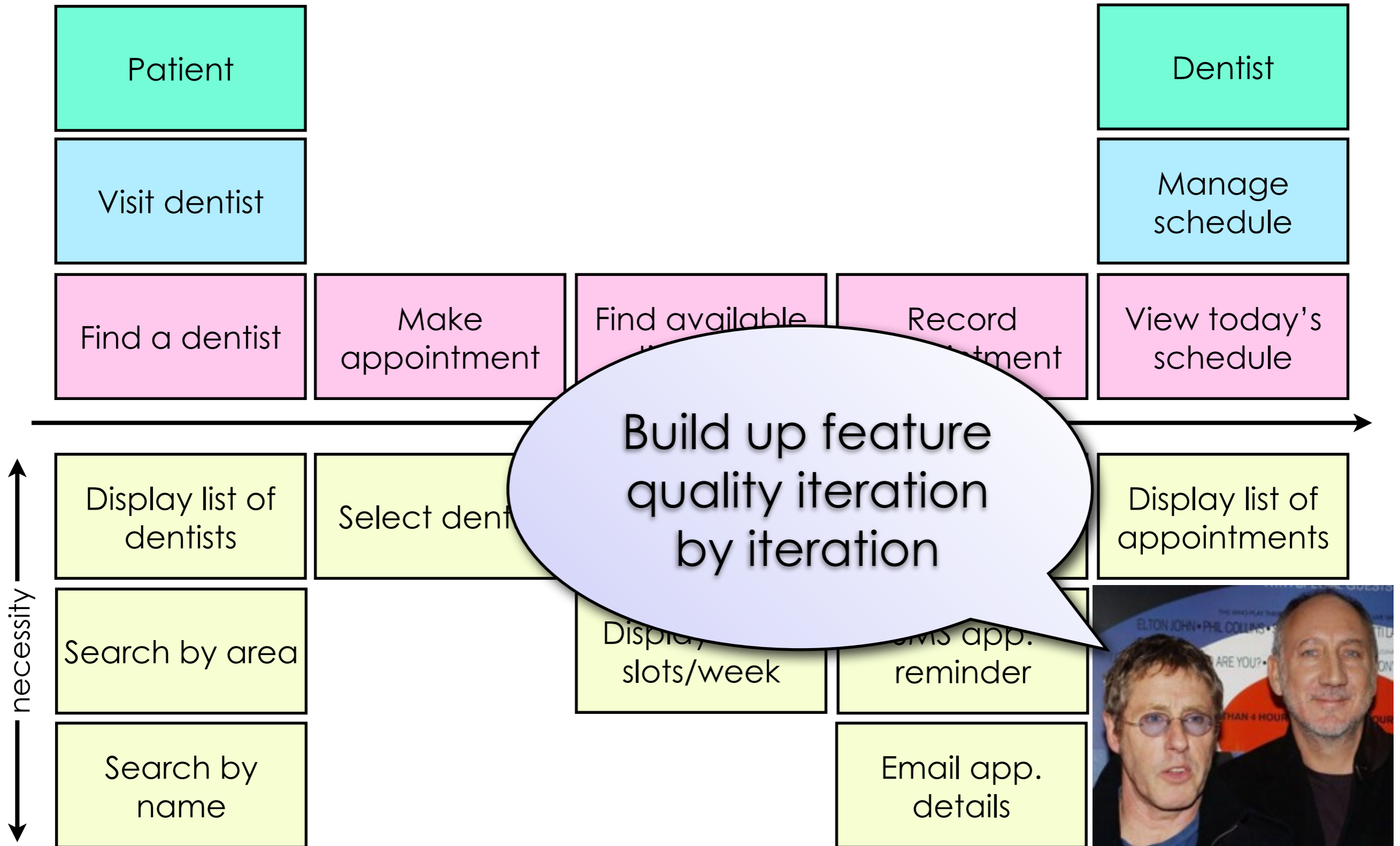
# Order features by necessity



# Order features by necessity

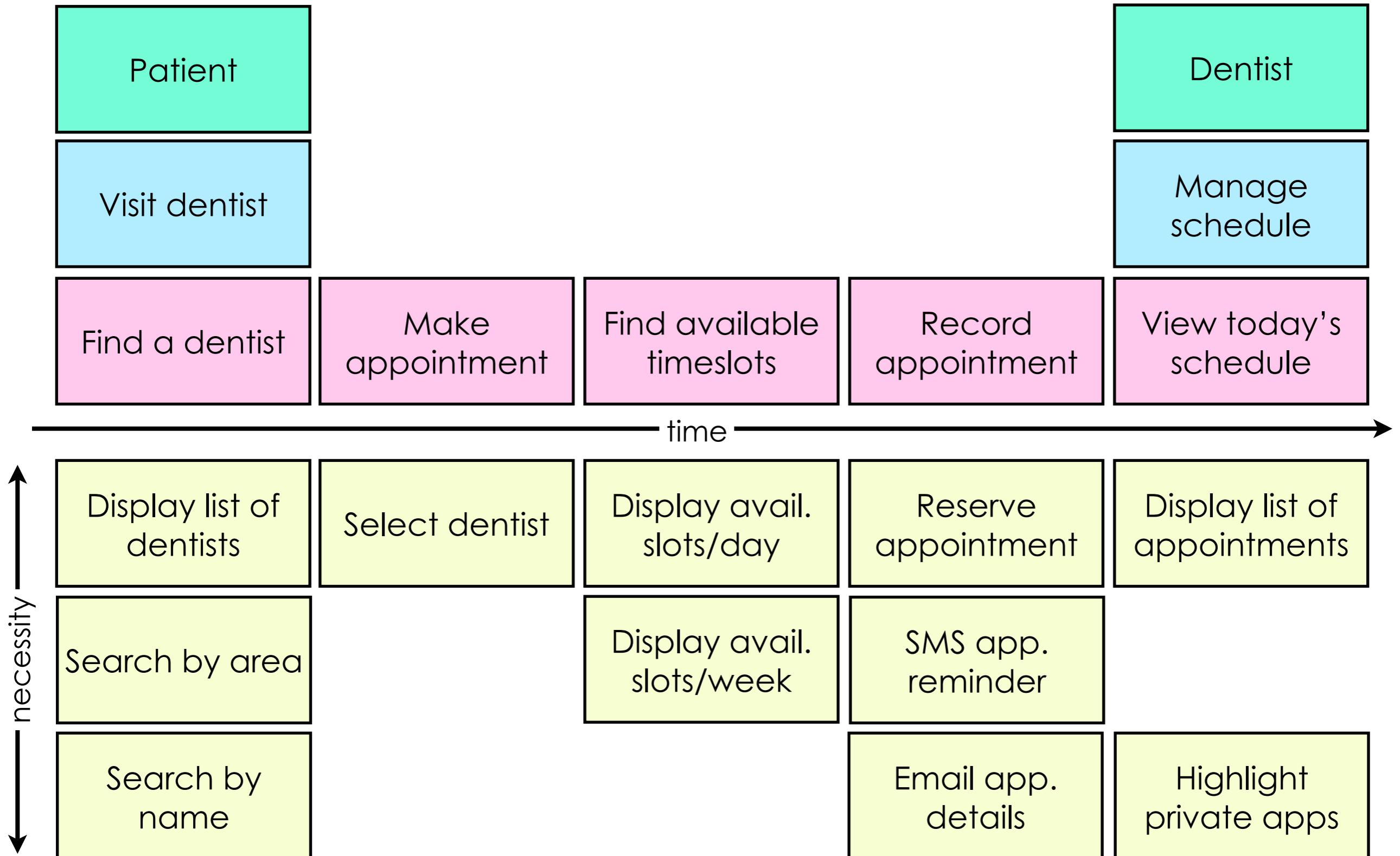


# Order features by necessity

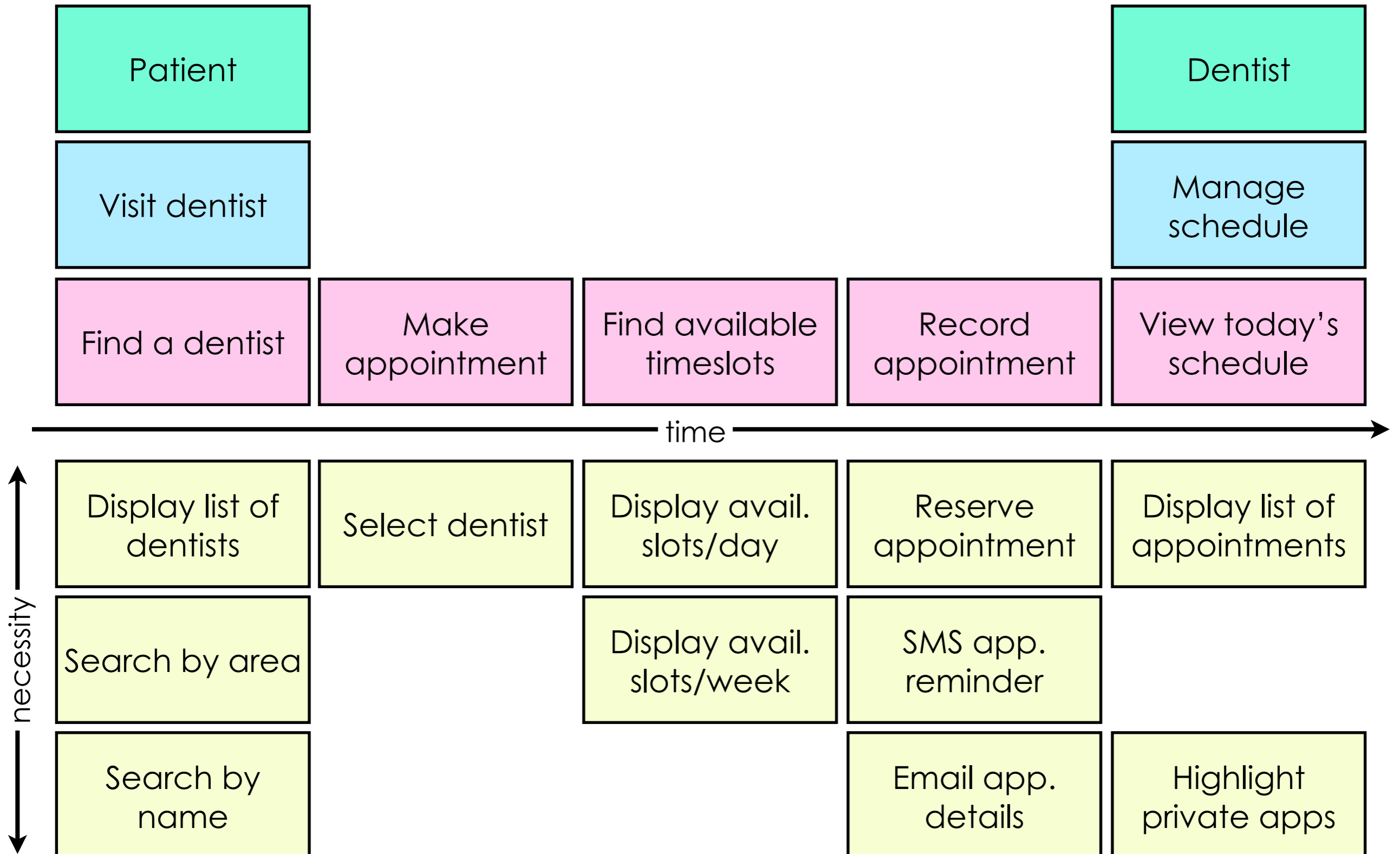




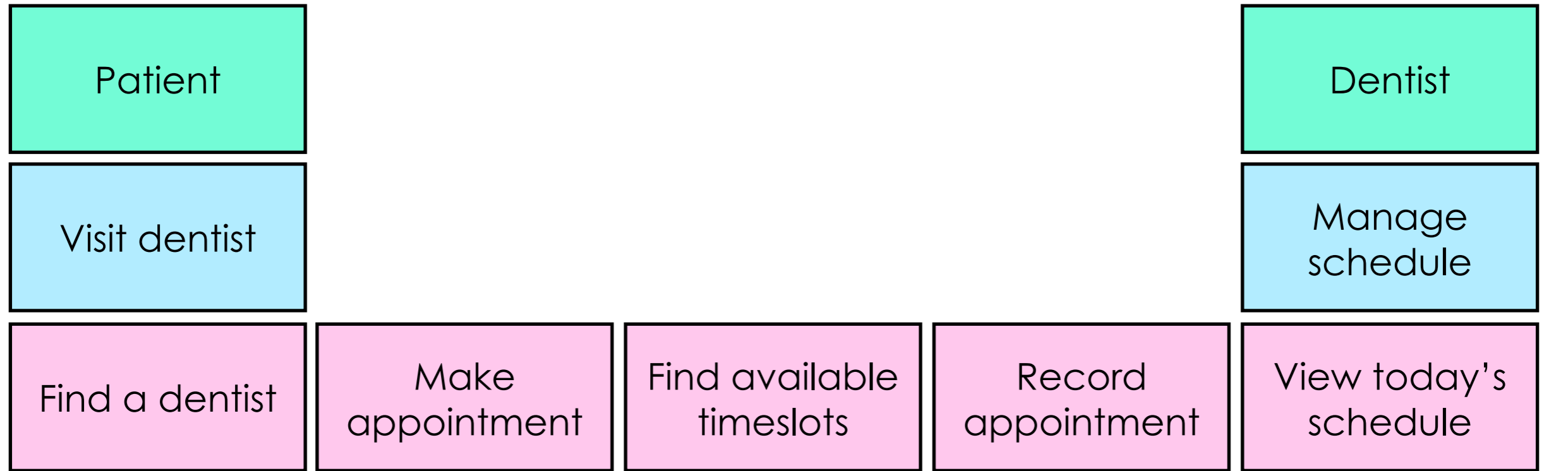
# Order features by necessity



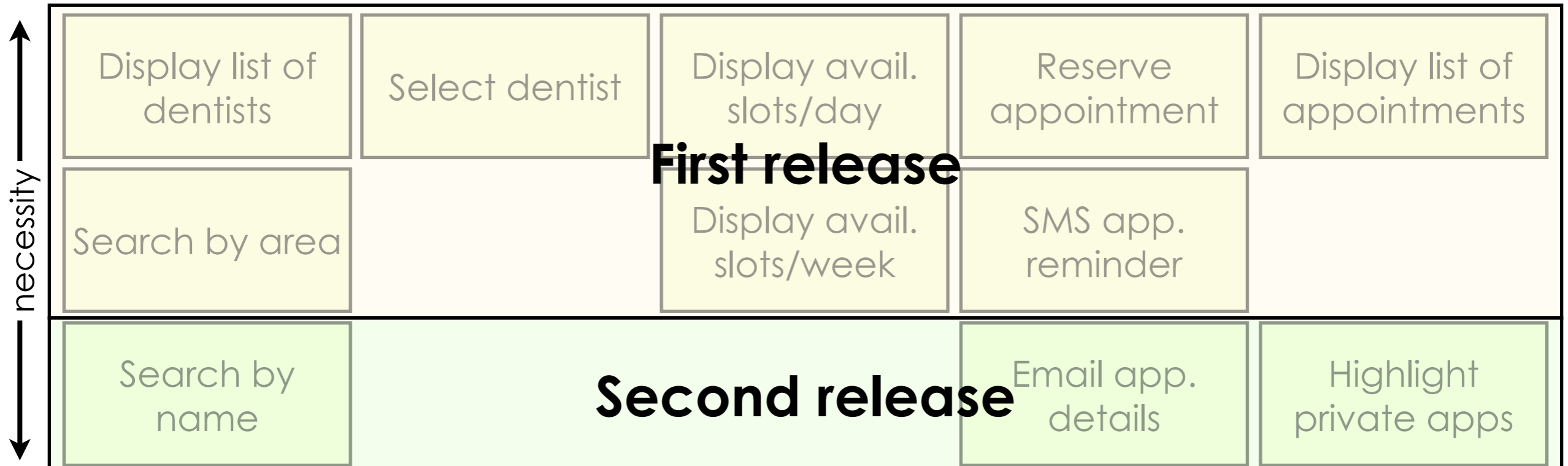
# Plan releases



# Plan releases



time →



**First release**

**Second release**



Let's review what our characters  
have learned today.





Roger now understands iteration and is a bit more cautious when interpreting his burn-down chart



Roger now understands iteration and is a bit more cautious when interpreting his burn-down chart

He knows that building software isn't like building a wall





Roger now understands iteration and is a bit more cautious when interpreting his burn-down chart

He knows that building software isn't like building a wall



Melanie thinks twice before asking what people want



Roger now understands iteration and is a bit more cautious when interpreting his burn-down chart

He knows that building software isn't like building a wall



Melanie thinks twice before asking what people want

She pays closer attention to their objectives

Roger leverages these three strategies for managing uncertainty:

Roger leverages these three strategies for managing uncertainty:



## Follow the Money

Roger leverages these three strategies for managing uncertainty:



Follow the *Money*



Don't choose your solution too early

Roger leverages these three strategies for managing uncertainty:



Follow the *Money*

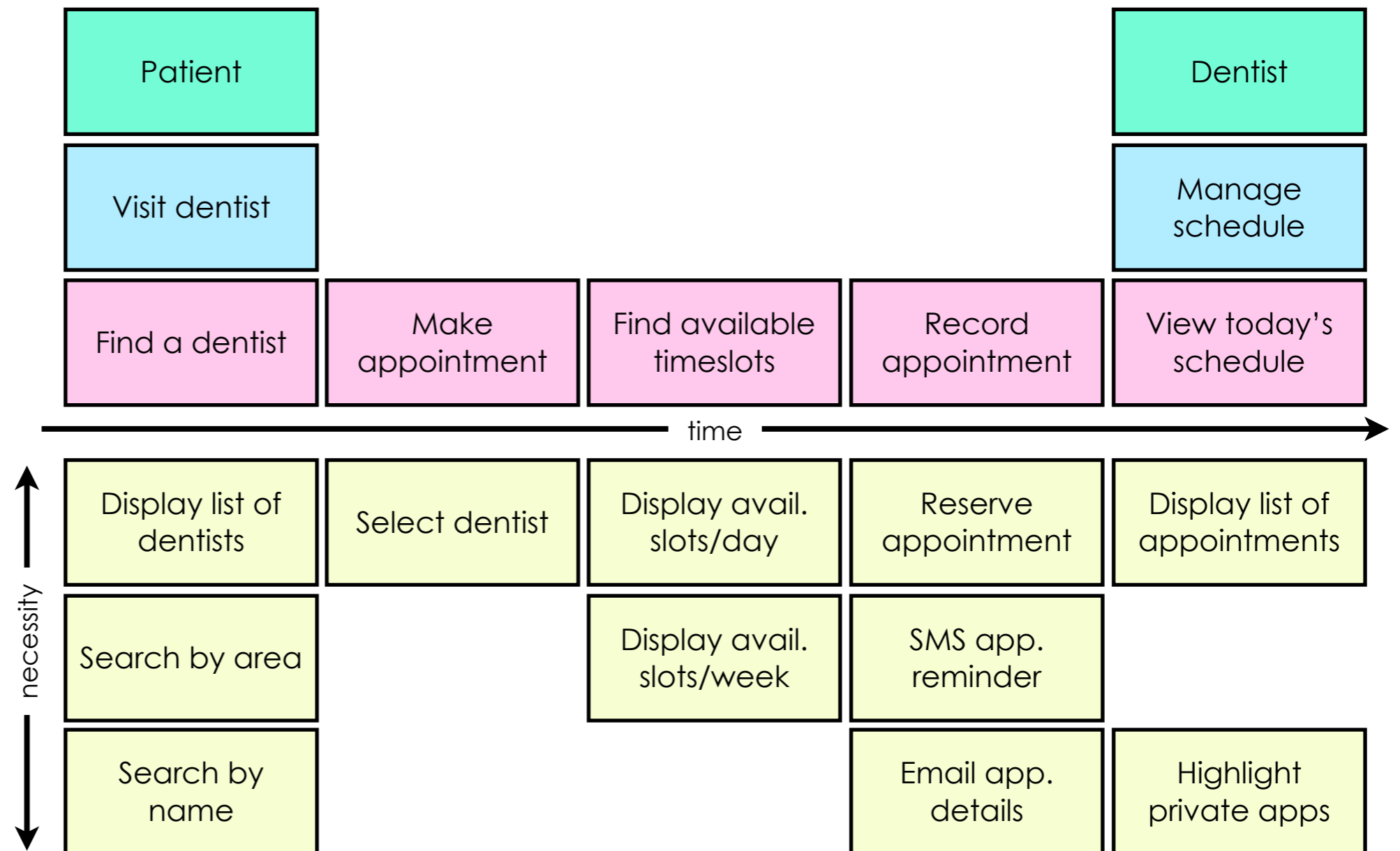


Don't choose your solution too early

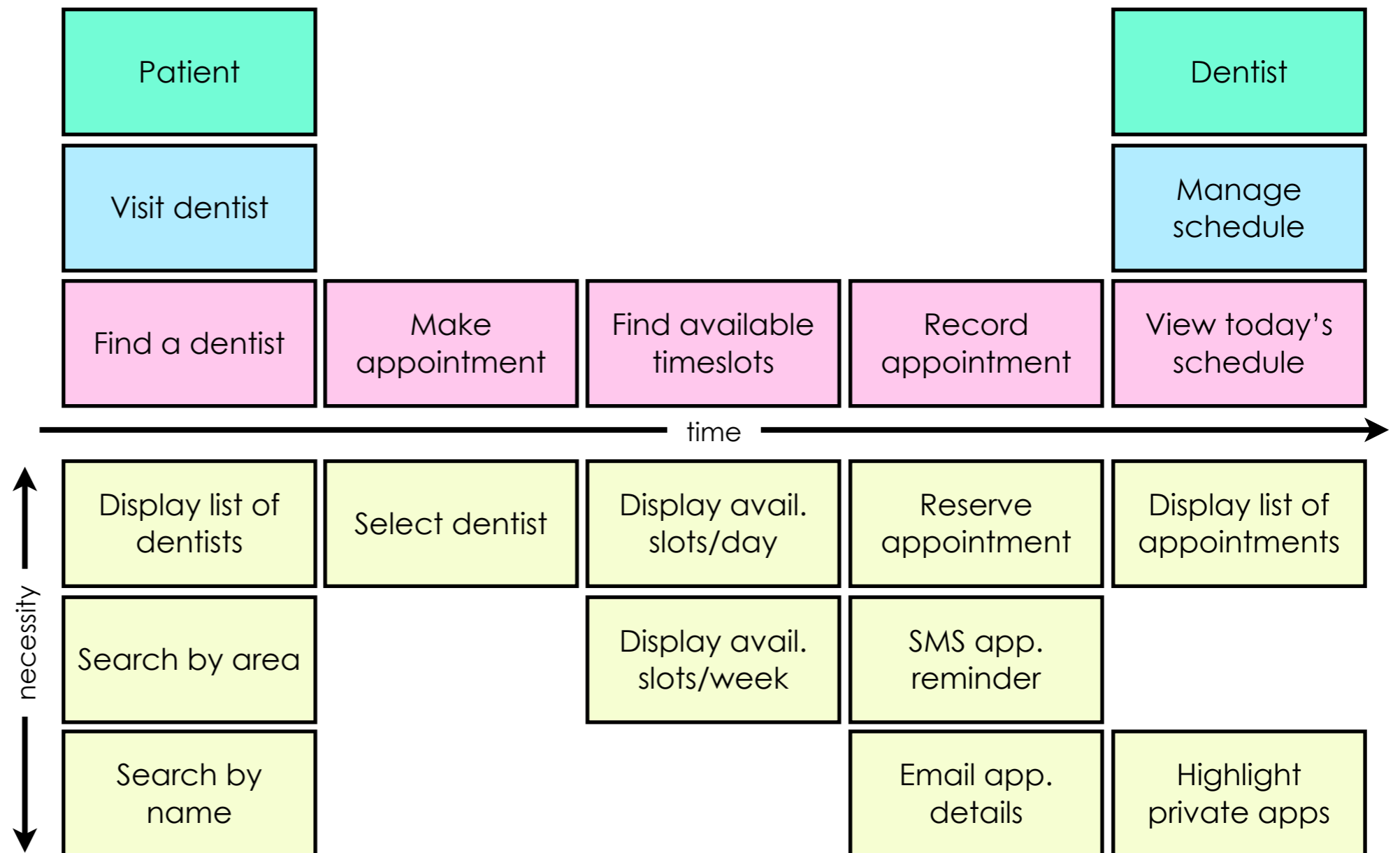


Build up feature quality iteration by iteration

Roger and Melanie leverage tools like User Story Mapping to implement the three strategies:



# Roger and Melanie leverage tools like User Story Mapping to implement the three strategies:







It's best not to be too certain about  
specifically what you're building

It's best not to be too certain about  
specifically what you're building

It's Johnny's wisdom that speaks  
clearly to this...





In the end, it's Johnny's approach that pays off when using XP and Agile development...

“Don't know what I want, but I know how to get it.”

# More information

# More information

Jeff Patton

<http://agileproductdesign.com/>

▶ [blog/dont\\_know\\_what\\_i\\_want.html](http://agileproductdesign.com/blog/dont_know_what_i_want.html)

# More information

Jeff Patton

<http://agileproductdesign.com/>

- ▶ [blog/dont\\_know\\_what\\_i\\_want.html](http://agileproductdesign.com/blog/dont_know_what_i_want.html)
- ▶ [blog/the\\_new\\_backlog.html](http://agileproductdesign.com/blog/the_new_backlog.html)



# More information

Jeff Patton

<http://agileproductdesign.com/>

- ▶ [blog/dont\\_know\\_what\\_i\\_want.html](http://agileproductdesign.com/blog/dont_know_what_i_want.html)
- ▶ [blog/the\\_new\\_backlog.html](http://agileproductdesign.com/blog/the_new_backlog.html)
- ▶ [presentations/user\\_story\\_mapping/](http://agileproductdesign.com/presentations/user_story_mapping/)

# User Story Mapping

and three strategies for managing uncertainty

Nils Christian Haugen  
nch@wasteless.no

wasteless