The Power of Unknowns

Harnessing what you don't know to achieve project success

John Keklak
Software Development Institute
john@keklak.com
About me

+ Software developer for >30 years
About me

+ Software developer for >30 years
+ Taught software engineering at BU for 9 years
About me

+ Software developer for >30 years
+ Taught software engineering at BU for 9 years
+ Still do a lot of software development
About me

+ Software developer for >30 years
+ Taught software engineering at BU for 9 years
+ Still do a lot of software development
+ Privately mentor aspiring software developers
Boston University
CS411 Software Engineering
BU CS411 Project Success

John Keklak
Software Development Institute
john@keklak.com
BU CS411 Project Success

The early years

John Keklak
Software Development Institute
john@keklak.com
BU CS411 Project Success

Later years...

John Keklak
Software Development Institute
john@keklak.com
BU CS411 Project Success

Later years…

What changed?

John Keklak
Software Development Institute
john@keklak.com
A story and a flash of insight

John Keklak
Software Development Institute
john@keklak.com
"They should have been doing this months ago!"

John Keklak
Software Development Institute
john@keklak.com
Doing what?

John Keklak
Software Development Institute
john@keklak.com
Doing what?

+ Identifying things they didn't know.
Doing what?

+ Identifying things they didn't know.
+ Resolving these things.
Doing what?

+ Identifying things they didn't know.
+ Resolving these things.

Well ahead of time.
Identifying and resolving things such as:
Identifying and resolving things such as:

+ How to get a browser to send data to a server without refreshing
Identifying and resolving things such as:

+ How to get a browser to send data to a server without refreshing
+ How to send an email with a server script
Identifying and resolving things such as:

+ How to get a browser to send data to a server without refreshing
+ How to send an email with a server script
+ What the existing product code did
Identifying and resolving things such as:

+ How to get a browser to send data to a server without refreshing
+ How to send an email with a server script
+ What the existing product code did
+ What the customer (me) really wanted...
Identifying and resolving things such as:

+ How to get a browser to send data to a server without refreshing
+ How to send an email with a server script
+ What the existing product code did
+ What the customer (me) really wanted...
+ Many other things...
A key concept fell into place for me
A key concept fell into place

+ What the customer wants
A key concept fell into place

+ What the customer wants
+ What the existing code does
A key concept fell into place

+ What the customer wants
+ What the existing code does
+ How to do various technical things
A key concept fell into place

+ What the customer wants
+ What the existing code does
+ How to do various technical things
+ What the UI should look like
A key concept fell into place

+ What the customer wants
+ What the existing code does
+ How to do various technical things
+ What the UI should look like

are... (drumroll)
A key concept fell into place

+ What the customer wants
+ What the existing code does
+ How to do various technical things
+ What the UI should look like

UNKNOWN!
Most likely, your reaction is...
Most likely, your reaction is...

SO WHAT...
Here's what...
Here's what...

Let's do a thought experiment...
A Thought Experiment

Situation 1: You are a software developer.
A Thought Experiment

Situation 1: You are a software developer. You are assigned to make some software product changes.
A Thought Experiment

Situation 1: You are a software developer. You are assigned to make some software product changes. You know nothing about the code.

John Keklak
Software Development Institute
john@keklak.com
A Thought Experiment

Situation 1: You are a software developer. You are assigned to make some software product changes. You know nothing about the code. You know vaguely about what the customer wants.
A Thought Experiment

Situation 1: You are a software developer. You are assigned to make some software product changes. You know nothing about the code. You know vaguely about what the customer wants. *You know little about the technology you will need to use.*
Situation 1: You are a software developer. You are assigned to make some software product changes. You know nothing about the code. You know vaguely about what the customer wants. You know little about the technology you will need to use. You have no idea what the UI should look like.
A Thought Experiment

Situation 1: You are a software developer. You are assigned to make some software product changes. You know nothing about the code. You know vaguely about what the customer wants. You know little about the technology you will need to use. You have no idea what the UI should look like.

How likely is it that this project will end successfully?

John Keklak                                          Software Development Institute                                          john@keklak.com
A Thought Experiment

Situation 2: You are a software developer.

John Keklak
Software Development Institute
john@keklak.com
Situation 2: You are a software developer. You are assigned to make some software product changes.
A Thought Experiment

Situation 2: You are a software developer. You are assigned to make some software product changes. You are fluent with the code.

John Keklak
Software Development Institute
john@keklak.com
A Thought Experiment

Situation 2: You are a software developer. You are assigned to make some software product changes. You are fluent with the code. **You know pretty clearly what the customer wants.**
Situation 2: You are a software developer. You are assigned to make some software product changes. You are fluent with the code. You know pretty clearly what the customer wants. You have experience with the technology you will need to use.
A Thought Experiment

Situation 2: You are a software developer. You are assigned to make some software product changes. You are fluent with the code. You know pretty clearly what the customer wants. You have experience with the technology you will need to use. A UI/UX colleague provides you with a pixel-perfect UI design.
Situation 2: You are a software developer. You are assigned to make some software product changes. You are fluent with the code. You know pretty clearly what the customer wants. You have experience with the technology you will need to use. A UI/UX colleague provides you with a pixel-perfect UI design.

How likely is it that this project will end successfully?
A Thought Experiment

Situation 1 vs Situation 2

What's different?
A Thought Experiment

Situation 1 vs Situation 2

What's different?

The amount of knowledge.
A Thought Experiment

Situation 1 vs Situation 2

*If we're in Situation 1,*
A Thought Experiment

Situation 1 vs Situation 2

If we're in Situation 1, and we want to get to Situation 2,
A Thought Experiment

Situation 1 vs Situation 2

If we're in Situation 1, and we want to get to Situation 2, how do we get there?
How do we get to Situation 2?

We need to gather knowledge.
So how do we get there?

We need to gather knowledge.

*What knowledge?*
So how do we get there?

We need to gather knowledge.
What knowledge?
The knowledge to carry out the project.
So how do we get there?

We need to gather knowledge.
What knowledge?
The knowledge to carry out the project.
Isn't that obvious?

John Keklak
Software Development Institute
john@keklak.com
So how do we get there?
We need to gather knowledge.
What knowledge? The knowledge to carry out the project. Isn't that obvious?

In many cases, no. Ask my students.
B.U. (before unknowns)
B.U. (*before unknowns*)

Get project assignment

John Keklak
Software Development Institute
john@keklak.com
B.U. (before unknowns)

Get project assignment

↓

Write code

John Keklak

Software Development Institute

john@keklak.com
B.U. (before unknowns)

Get project assignment

Write code, **whack and hack**

John Keklak                            Software Development Institute                                  john@keklak.com
B.U. (before unknowns)

Get project assignment

Write code, whack and hack

Usually fail

John Keklak
Software Development Institute
john@keklak.com
Get project assignment
A.U. (after unknowns)

Get project assignment

Identify what you don't know
A.U. (after unknowns)

Get project assignment

Identify what you don't know
(there's lots; be as thorough as you can)
A.U. (after unknowns)

Get project assignment

Identify what you don't know
(there's lots; be as thorough as you can)

Gather knowledge
A.U. (after unknowns)

Get project assignment

↓

Identify what you don't know
(there's lots; be as thorough as you can)

↓

Gather knowledge
(involves writing lots of code)
A.U. (after unknowns)

1. Get project assignment
2. Identify what you don't know
   (there's lots; be as thorough as you can)
3. Gather knowledge
   (involves writing lots of code)
4. Write product code
A.U. (after unknowns)

Get project assignment

Identify what you don't know
(there's lots; be as thorough as you can)

Gather knowledge
(involves writing lots of code)

Write product code
(quickly and well)
Write product code
(quickly and well)
Write product code (quickly and well)

How do I know?
Write product code
(quickly and well)

How do I know?

Another story (actually two)
Write product code
(quickly and well)

How do I know?

Story #1

"It is easier to apologize than to ask for permission."

John Keklak
Software Development Institute
john@keklak.com
Write product code
(quickly and well)

How do I know?

Story #2

"What happens when your command line syntax is incorrect."

John Keklak
Software Development Institute
john@keklak.com
Write product code
(quickly and well)
A.U. (after unknowns)

Get project assignment

Identify what you don't know
(there's lots; be as thorough as you can)

Gather knowledge
(involves writing lots of code)

Write product code
(quickly and well)
What does it mean to "gather knowledge"?
What does it mean to "gather knowledge"?

Involves writing lots of code;

John Keklak  Software Development Institute  john@keklak.com
What does it mean to "gather knowledge"?

Involves writing lots of code; creating lots of little programs

John Keklak
Software Development Institute
john@keklak.com
What does it mean to "gather knowledge"?

Involves writing lots of code; creating lots of little programs
What does it mean to "gather knowledge"?

Involves writing lots of code; creating lots of little programs

Talking to customers

Iterating through UI designs

John Keklak  Software Development Institute  john@keklak.com
What does it mean to "gather knowledge"?

- Involves writing lots of code; creating lots of little programs
- Talking to customers
- Iterating through UI designs
- Getting fluent with existing code
What does it mean to "gather knowledge"?

- Involves writing lots of code; creating lots of little programs.
- Talking to customers.
- Iterating through UI designs.
- Getting fluent with existing code.

***Don't worry about writing final product code (yet)***
What does it mean to "gather knowledge"?

Involves writing lots of code; creating lots of little programs

Talking to customers

Iterating through UI designs

Getting fluent with existing code

***Don't worry about writing final product code (yet)***

Expect to spend lots of time here
What does it mean to "gather knowledge"?

Involves writing lots of code; creating lots of little programs
A.U. (after unknowns)

Get project assignment

Identify what you don't know
(there's lots; be as thorough as you can)

Gather knowledge
(involves writing lots of code)

Write product code
(quickly and well)
A.U. (after unknowns)

Get project assignment

Identify what you don't know
(there's lots; be as thorough as you can)

Gather knowledge
(involves writing lots of code)

Write product code
(quickly and well)
What does it mean to "identify what you don't know"?

John Keklak  
Software Development Institute  
john@keklak.com
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

John Keklak  Software Development Institute  john@keklak.com
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.

It takes a lot of discipline.
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.

It takes a lot of discipline.

Developers always rush through this
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.

It takes a lot of discipline.

Developers always rush through this
(I know, I've rushed through this)
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.

It takes a lot of discipline.

Developers always rush through this (I know, I've rushed through this)

If you can't think of anything you don't know, start here:

John Keklak

Software Development Institute

john@keklak.com
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.

It takes a lot of discipline.

Developers always rush through this (I know, I've rushed through this)

If you can't think of anything you don't know, start here:

What does the customer want?
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.

It takes a lot of discipline.

Developers always rush through this (I know, I've rushed through this)

If you can't think of anything you don't know, start here:

What does the customer want?  What should the UI look like?

John Keklak                           Software Development Institute                                  john@keklak.com
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.

It takes a lot of discipline

Developers always rush through this
(I know, I've rushed through this)

If you can't think of anything you don't know, start here:

What does the customer want?  What should the UI look like?

How does existing code work?
What does it mean to "identify what you don't know"?

Well, identify what you don't know.

Make a list.

There's lots; be as thorough as you can.

It takes a lot of discipline

Developers always rush through this
(I know, I've rushed through this)

If you can't think of anything you don't know, start here:

What does the customer want? What should the UI look like?
How does existing code work? How will required features work?
Identify what you don't know

A helpful way to phrase unknowns:
Identify what you don't know

A helpful way to phrase unknowns:

"What I don't know is…"
Some concrete examples of unknowns
Some concrete examples of unknowns

General:

"What I don't know is... what the customer wants."
Some concrete examples of unknowns

General:
"What I don't know is... what the customer wants."

Specific:
"What I don't know is... how the customer wants account registration to work."
Some concrete examples of unknowns

General:

"What I don't know is... what the UI should look like."
Some concrete examples of unknowns

General:
“What I don't know is... what the UI should look like.”

Specific:
“What I don't know is… what the UI for account registration should look like.”
Some concrete examples of unknowns

General:

"What I don't know is... the overall architecture to use."
Some concrete examples of unknowns

General:
"What I don't know is...
the overall architecture to use."

Specific:
"What I don't know is…
how to get a browser to send data to a server without refreshing."
Some concrete examples of unknowns

Specific:

"What I don't know is…
how to send an email with a server script."
Some concrete examples of unknowns

Specific:

"What I don't know is… how to send an email with a server script."

"What I don't know is… how the existing code for logging in works."
Rule of thumb for unknowns

John Keklak
Software Development Institute
john@keklak.com
Rule of thumb for unknowns

Specific is better than general.

John Keklak
Software Development Institute
john@keklak.com
Rule of thumb for unknowns

Specific is better than general.

General:

"What I don't know is... what the customer wants."

John Keklak
Software Development Institute
john@keklak.com
Rule of thumb for unknowns

Specific is better than general.

General:
"What I don't know is... what the customer wants."

Specific:
"What I don't know is… how the customer wants account registration to work."

John Keklak  Software Development Institute  john@keklak.com
Another rule of thumb
Another rule of thumb

Create a comprehensive list before you start to resolve unknowns.
Another rule of thumb

Create a comprehensive list before you start to resolve unknowns.

If you don't have pages of unknowns, you haven't been thorough enough.
Resolving unknowns

John Keklak
Software Development Institute
john@keklak.com
Resolving unknowns

= gathering knowledge
Resolving unknowns

Unknown:

"What I don't know is…
what the customer wants."
Resolving unknowns

Unknown:
"What I don't know is... what the customer wants."

Solution:

Discuss the project with the customer.
Resolving unknowns

Unknown:

"What I don't know is… how the customer wants account registration to work."

John Keklak                            Software Development Institute                                  john@keklak.com
Resolving unknowns

Unknown:
"What I don't know is... how the customer wants account registration to work."

Solution:
Discuss account registration with the customer.
Resolving unknowns

Unknown:
"What I don't know is... how the customer wants account registration to work."

Solution:
Discuss account registration with the customer.
Write scenarios.
Resolving unknowns

Unknown:
"What I don't know is...
how the customer wants account registration to work."

Solution:
Discuss account registration with the customer. Write scenarios. Review, revise, repeat.
Resolving unknowns

Unknown:

"What I don't know is... what the UI for account registration should look like."
Resolving unknowns

Unknown:

“What I don't know is... what the UI for account registration should look like.”

Solution:

Create a paper UI.

John Keklak
Software Development Institute
john@keklak.com
Resolving unknowns

Unknown:
"What I don't know is... what the UI for account registration should look like."

Solution:
Create a paper UI. Review it with the customer and/or colleagues.
Resolving unknowns

Unknown:

"What I don't know is... what the UI for account registration should look like."

Solution:

Create a paper UI. Review it with the customer and/or colleagues.

Revise the paper UI.
Resolving unknowns

Unknown:

"What I don't know is... what the UI for account registration should look like."

Solution:

Create a paper UI. Review it with the customer and/or colleagues. Revise the paper UI. Repeat.
Resolving unknowns

Unknown:

"What I don't know is how the existing code for logging in works."
Resolving unknowns

Unknown:

"What I don't know is... how the existing code for logging in works."

Solution:

Study it.

John Keklak
Software Development Institute
john@keklak.com
Resolving unknowns

Unknown:

"What I don't know is... how the existing code for logging in works."

Solution:

Study it.  *Discuss it with someone knowledgable.*
Resolving unknowns

Unknown:

"What I don't know is... how the existing code for logging in works."

Solution:

Study it. Discuss it with someone knowledgable. 

Single-step through the code.
Resolving unknowns

Unknown:

"What I don't know is... how the existing code for logging in works."

Solution:

Study it. Discuss it with someone knowledgable. Single-step through the code. Refactor it.

John Keklak
Software Development Institute
john@keklak.com
Resolving unknowns

Unknown:

"What I don't know is… how to send email with a server script."

John Keklak

Software Development Institute

john@keklak.com
Resolving unknowns

Unknown:
"What I don't know is... how to send email with a server script."

Solution:
Google for information.

John Keklak
Software Development Institute
john@keklak.com
Resolving unknowns

Unknown:
"What I don't know is... how to send email with a server script."

Solution:

Google for information. **Write a small script that successfully sends an email.**
Resolving unknowns

Unknown:
"What I don't know is... how to send email with a server script."

Solution:

Google for information. Write a small script that successfully sends an email. **Write up key points.**

John Keklak                                  Software Development Institute                                  john@keklak.com
Explore all the unknowns on your list.
Does this technique scale?
Do I use this technique for my own projects?
Do I use this technique for my own projects?

Always.

John Keklak
Software Development Institute
john@keklak.com
In conclusion...
B.U.
A.U.

![Smiley faces](image_url)
A.U.

Focusing students on unknowns made all the difference.

John Keklak
Software Development Institute
john@keklak.com
Other benefits we don't have time to get into tonight.
Other benefits we don't have time to get into tonight.

+ Much easier estimating and scheduling
Other benefits we don't have time to get into tonight.

+ Much easier estimating and scheduling

+ Much more effective communication
Other benefits we don't have time to get into tonight.

+ Much easier estimating and scheduling
+ Much more effective communication between developers and team leaders
Other benefits we don't have time to get into tonight.

+ Much easier estimating and scheduling
+ Much more effective communication between developers and team leaders
  between developers and customers
Other benefits we don't have time to get into tonight.

+ Much easier estimating and scheduling
+ Much more effective communication between developers and team leaders between developers and customers between developers and developers
Other benefits we don't have time to get into tonight.

+ Much easier estimating and scheduling

+ Much more effective communication
  between developers and team leaders
  between developers and customers
  between developers and developers

+ Developers embrace the approach
Other benefits we don't have time to get into tonight.

+ Much easier estimating and scheduling
+ Much more effective communication between developers and team leaders
  between developers and customers
  between developers and developers
+ Developers embrace the approach (this cannot be understated)
Benefits we don't have time to get into tonight.

+ Much easier estimating and scheduling

+ Much more effective communication all around

+ Developers embrace the approach

For more information, contact me

John Keklak

Software Development Institute

john@keklak.com
Thank you for your time and attention!
The Power of Unknowns

Questions?

John Keklak  Software Development Institute  john@keklak.com