

29 TECHNIQUES FOR REDUCING THE COGNITIVE LOAD

that you won't hear
them talking about



Alexey
Krivitsky

Org Consultant
De/Scaling Expert
Product & Engineering Coach

who is standing on the shoulders of the giants



> How it all began ...



Viktor Grgic • 1st
Software Product development expert - Certified LeSS trainer - he/him
5mo • 🗨️

...

The lack of whole product focus and proximity to customers is another common reason for increased teams' cognitive load. The mistaken assumption about the load is that teams should not be involved in product discovery so "tl ...see more

👍👍👍 31 7 comments • 3 reposts



Viktor Grgic • 1st
Software Product development expert - Certified LeSS trainer - he/him
5mo • Edited • 🗨️

Another way to reduce teams' cognitive load is to avoid having teams coordinating and chasing project managers and fake POs (responsible for coordination). Instead, skip any intermediary and have teams coordi ...see more

👍👍👍 39 16 comments



Viktor Grgic • 1st
Software Product development expert - Certified LeSS trainer - he/him
5mo • Edited • 🗨️

...

Yet another way to reduce teams' cognitive load is by improving *testing automation*. Teams with high confidence that changes will not cause trouble in production if all tests are green can handle more components and d ...see more

👍👍👍 33 1 comment • 2 reposts



Viktor Grgic • 1st
Software Product development expert - Certified LeSS trainer - he/him
5mo • Edited • 🗨️

...

Yet another way to reduce a team's cognitive load is to avoid spreading team members across different locations (especially time zones). Dispersed team members will have scattered knowledge, which is wasteful. This pag ...see more

👍👍 17 15 comments



Viktor Grgic • 1st
Software Product development expert - Certified LeSS trainer - he/him
5mo • 🗨️

...

Another great way to reduce teams' cognitive load. TDD is about taking small steps when solving complex problems.

...see more



Kent Beck (He/Him) • 1st
Programmer, artist, musician, pokerist
5mo • 🗨️

..

Canon TDD -- I keep reading critiques of processes that aren't TDD, so I summarized the original definition.



Viktor Grgic • 1st
Software Product development expert - Certified LeSS trainer - he/him
5mo • 🗨️

...

Another way to reduce the team's cognitive load:
Avoid having individual team members own "tickets." Sharing the work improves focus, speeds delivery, and spreads the load.
A full-stack developer practice is quite popular, but it is often misunderstood as having individual devs doing all activities to solve a customer problem by themselves. The assumption is that people learn a lot by trying to achieve this ability. In reality, there is mostly pain, pressure, and stress to deliver something alone quickly.

#lessworks

👤 Dirk Lässig and 30 others 10 comments • 4 reposts



Viktor Grgic • 1st
Software Product development expert - Certified LeSS trainer - he/him
5mo • 🗨️

...

Do you want to reduce teams' cognitive load? This is one of the many ways to reduce cognitive load.

...see more



Andree Surya • 2nd
Technical Lead at DKATALIS (Digital Katalis)
1yr • 🗨️

+ Follow

Not just a mere machine instruction—People often forget that software source code is a living specification. You are very likely to re-read your source code from time-to-time to confirm its behavior, fix it, or ...see more

> “Ah! There is more than *one* way ...”

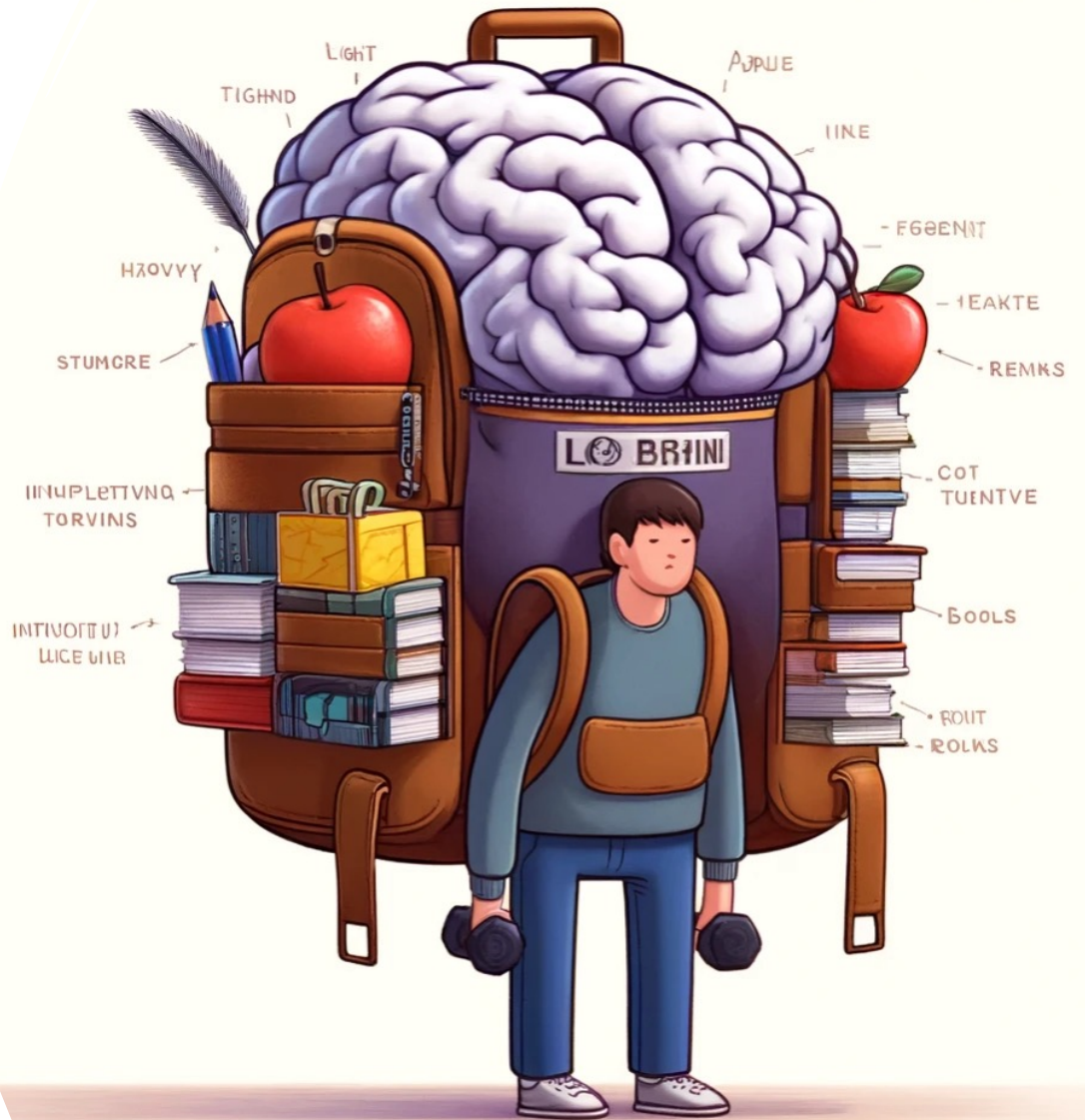
Following Yesterday's Advice from Craig ...

We'll start with the right understanding of Cognitive Load.



Cognitive Load

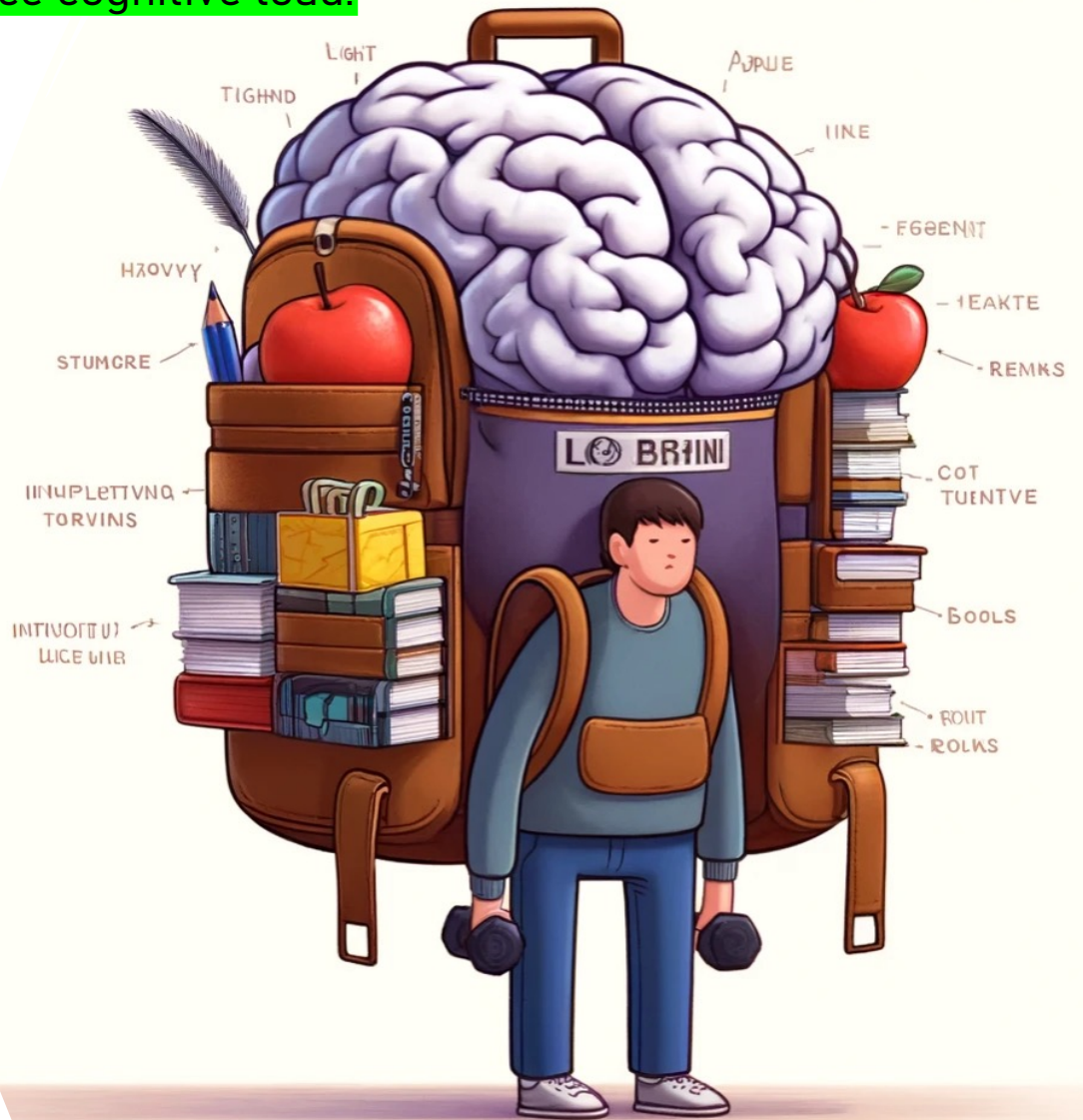
Imagine your brain is like a backpack that can carry information. Cognitive load is about how heavy the stuff you need to carry is.



This slide “hides” at least two techniques to reduce cognitive load.

Cognitive Load

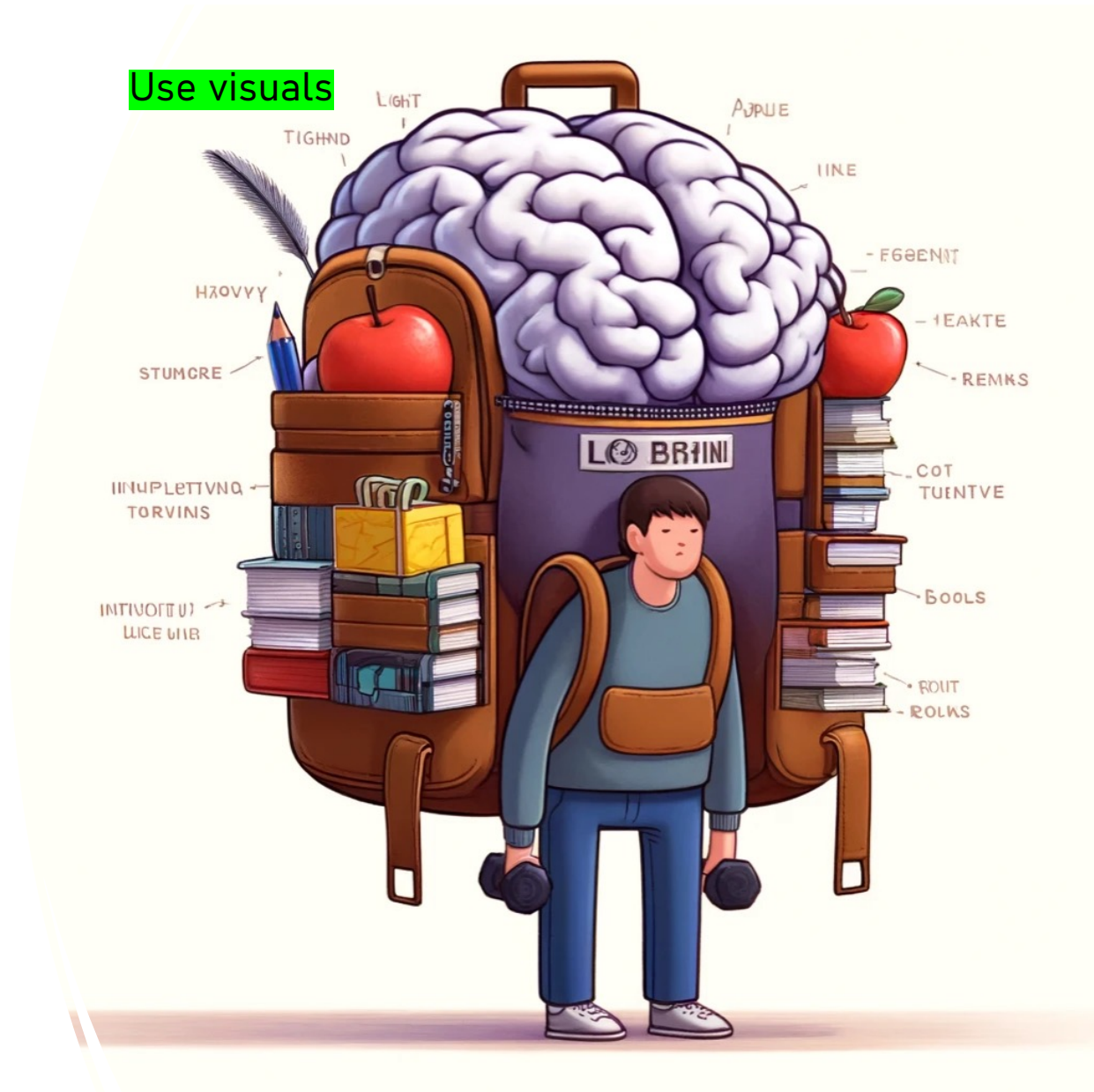
Imagine your brain is like a backpack that can carry information. Cognitive load is about how heavy the stuff you need to carry is.



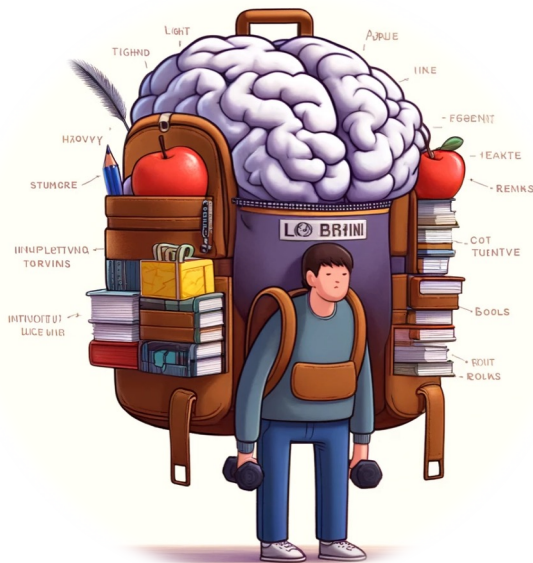
Cognitive Load

Imagine your brain is like a backpack that can carry information. Cognitive load is about how heavy the stuff you need to carry is.

Use metaphors and analogies



(1) Intrinsic



- is just a fancy way of saying how hard something is to learn or do because of how tricky or complicated it is.

(2) Extraneous



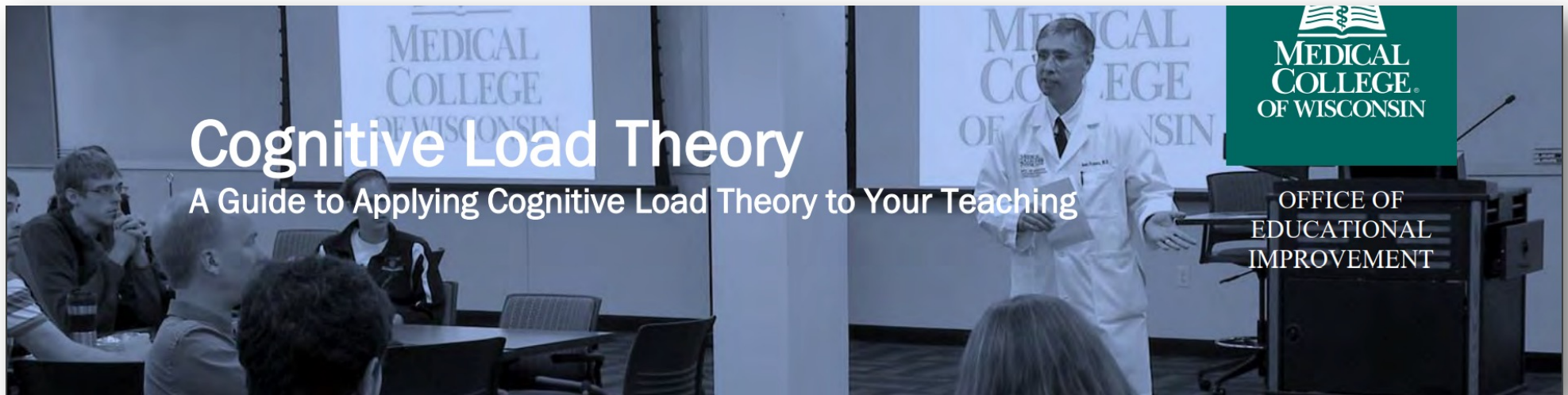
- is about things that make it harder to learn or do something but aren't actually part of what you need to learn.

(3) Germane



- is the mental effort you put into understanding, organizing, and integrating new information in a way that helps you learn better.

Cognitive Load Theory (CLT)



CLT is an “educational psychology”-centric science.

Medical College of Wisconsin, “Cognitive Load Theory”

mcw.edu/-/media/MCW/Education/Academic-Affairs/OEI/Faculty-Quick-Guides/Cognitive-Load-Theory.pdf

> CL is the effort to learn (i.e., understand, process and organize new information for further retrieval and usage).

> CL has nothing to do with how much information our brains can store!

These limits are not known to the science.

> RAM, not HDD.

```

public class BasketCalculator {

    // Inner class representing a Product
    public static class Product {
        private String name;
        private double price;
        private double taxRate;

        public Product(String name, double price, double taxRate) {
            this.name = name;
            this.price = price;
            this.taxRate = taxRate;
        }

        public double calculateTax() {
            return price * taxRate / 100;
        }

        public double getPriceWithTax() {
            return price + calculateTax();
        }

        public String getName() {
            return name;
        }

        public double getPrice() {
            return price;
        }

        public double getTaxRate() {
            return taxRate;
        }
    }

    // Method to calculate the total price of all items in the basket, including tax and discount
    public double calculateTotal(List<Product> products, String discountCode) {
        double totalPriceBeforeDiscount = 0.0;

        // Sum up the price with tax for each product in the basket
        for (Product product : products) {
            totalPriceBeforeDiscount += product.getPriceWithTax();
        }

        // Apply discount based on the discount code
        double discount = getDiscountPercentage(discountCode);
        double discountAmount = totalPriceBeforeDiscount * discount / 100;

        // Calculate the final total amount to be paid
        return totalPriceBeforeDiscount - discountAmount;
    }

    // Helper method to retrieve discount percentage based on discount code
    private double getDiscountPercentage(String discountCode) {
        // Simple discount code logic (you can add more logic as needed)
        if (discountCode == null || discountCode.isEmpty()) {
            return 0.0; // No discount if no code is provided
        }

        switch (discountCode.toLowerCase()) {
            case "save10":
                return 10.0; // 10% discount
            case "save20":
                return 20.0; // 20% discount
            case "save30":
                return 30.0; // 30% discount
            default:
                return 0.0; // Invalid or unknown code gives no discount
        }
    }
}

```

> Some clean code

```

-
  @registration.attributes = registration_params
rescue
end

@registration.participants.each do | EventParticipant p|
  p.event_registration = @registration
  p.event_occasion = @registration.event_occasion

  p.no_participant_data = @registration.no_participant_data
end

if @registration.self_participant
  owner = @registration.find_or_build_owner_participant
  owner.copy_from @registration
end

if params["event_registration"]["no_submit"] == "self_participant_changed"
  @registration.ignore_amo_sync = true

  if @registration.self_participant
    owner = @registration.find_or_build_owner_participant
    owner.copy_from @registration
    @registration.participant_count = @registration.event_participants.size
  else
    owner = @registration.find_or_build_owner_participant
    owner.destroy
    @registration.event_participants.delete(owner)
    if @registration.event_participants.empty?
      new_participant = @registration.event_participants.build
      new_participant.event_occasion = @registration.event_occasion
      new_participant.event_registration = @registration
    end
    @registration.participant_count = @registration.event_participants.size
  end

  @registration.prepare_all_pricing_stuff

  respond_to do |format|
    format.html { render :show }
    format.json { render :json, status: :ok }
  end
end

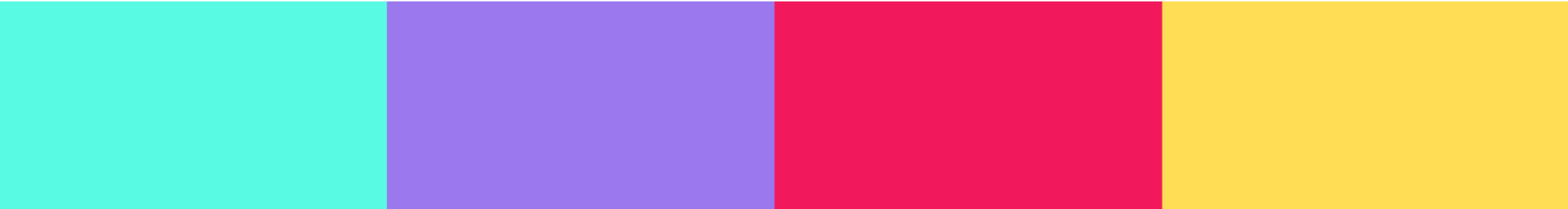
```

> Some **dirty code**

Would you prefer to work with
a lot of **clean code** or a little bit of **dirty code**?

```
public class BasketCalculator {  
    // Inner class representing a Product  
    public static class Product {  
        private String name;  
        private double price;  
        private double taxRate;  
  
        public Product(String name, double price, double taxRate) {  
            this.name = name;  
            this.price = price;  
            this.taxRate = taxRate;  
        }  
  
        public double calculateTax() {  
            return price * taxRate / 100;  
        }  
  
        public double getPriceWithTax() {  
            return price + calculateTax();  
        }  
  
        public String getName() {  
            return name;  
        }  
  
        public double getPrice() {  
            return price;  
        }  
  
        public double getTaxRate() {  
            return taxRate;  
        }  
    }  
  
    // Method to calculate the total price of all items in the basket, including tax and discount  
    public double calculateTotal(List<Product> products, String discountCode) {  
        double totalPriceBeforeDiscount = 0.0;  
  
        // Sum up the price with tax for each product in the basket  
        for (Product product : products) {  
            totalPriceBeforeDiscount += product.getPriceWithTax();  
        }  
  
        // Apply discount based on the discount code  
        double discount = getDiscountPercentage(discountCode);  
        double discountAmount = totalPriceBeforeDiscount * discount / 100;  
  
        // Calculate the final total amount to be paid  
        return totalPriceBeforeDiscount - discountAmount;  
    }  
  
    // Helper method to retrieve discount percentage based on discount code  
    private double getDiscountPercentage(String discountCode) {  
        // Simple discount code logic (you can add more logic as needed)  
        if (discountCode == null || discountCode.isEmpty()) {  
            return 0.0; // No discount if no code is provided  
        }  
  
        switch (discountCode.toLowerCase()) {  
            case "save10":  
                return 10.0; // 10% discount  
            case "save20":  
                return 20.0; // 20% discount  
            case "save30":  
                return 30.0; // 30% discount  
            default:  
                return 0.0; // Invalid or unknown code gives no discount  
        }  
    }  
}
```

```
@Registration.attributes = registration_params  
rescue  
end  
  
@registration.participants.each do |EventParticipant p|  
    p.event_registration = @registration  
    p.event_occasion = @registration.event_occasion  
  
    p.no_participant_data = @registration.no_participant_data  
end  
  
if @registration.self_participant  
    owner = @registration.find_or_build_owner_participant  
    owner.copy_from @registration  
end  
  
if params["event_registration"]["no_submit"] == "self_participant_changed"  
    @registration.ignore_emo_sync = true  
  
    if @registration.self_participant  
        owner = @registration.find_or_build_owner_participant  
        owner.copy_from @registration  
        @registration.participant_count = @registration.event_participants.size  
    else  
        owner = @registration.find_or_build_owner_participant  
        owner.destroy  
        @registration.event_participants.delete(owner)  
        if @registration.event_participants.empty?  
            new_participant = @registration.event_participants.build  
            new_participant.event_occasion = @registration.event_occasion  
            new_participant.event_registration = @registration  
        end  
        @registration.participant_count = @registration.event_participants.size  
    end  
  
    @registration.prepare_all_pricing_stuff  
  
    respond_to do |format|  
        # ...  
    end  
end
```



Limit the size of software services/products to the cognitive load that the team can handle.

JANUARY 21, 2021

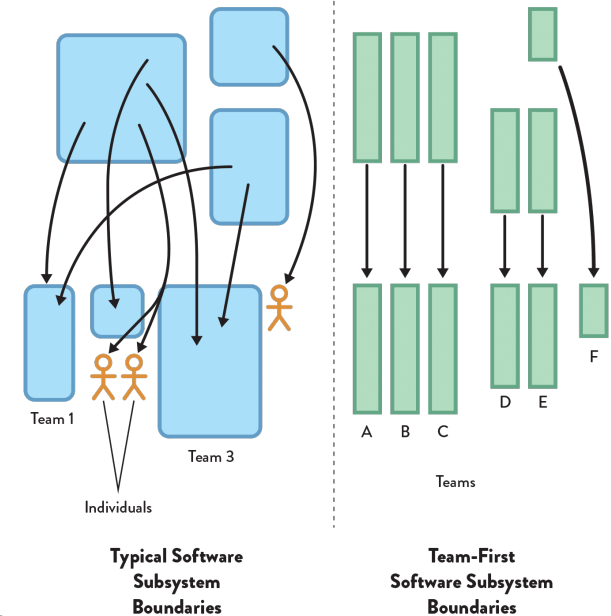
Minimize Team Cognitive Load to Increase Flow

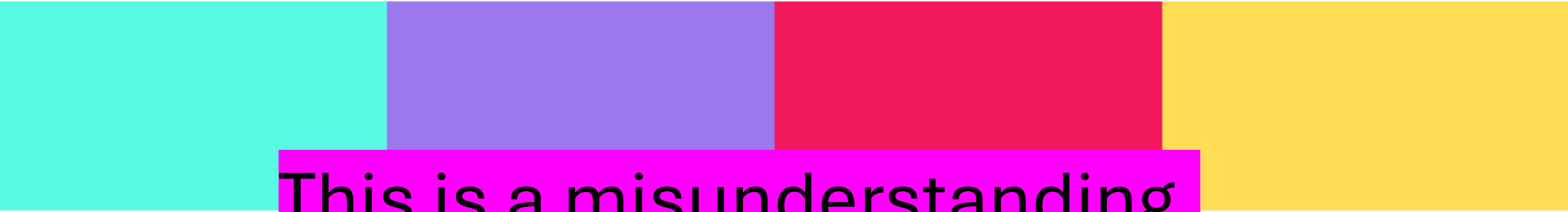
Heuristics of Domains per Team

- 1) Assign Each Domain a Team
- 2) A Single Team Should Accommodate 2-3 Simple Domains
- 3) Teams Responsible for Complex Domain Should Have No Additional Domains
- 4) Avoid Single Team Responsible for Two Complicated Domains

Match Software Boundary Size to Team Cognitive Load

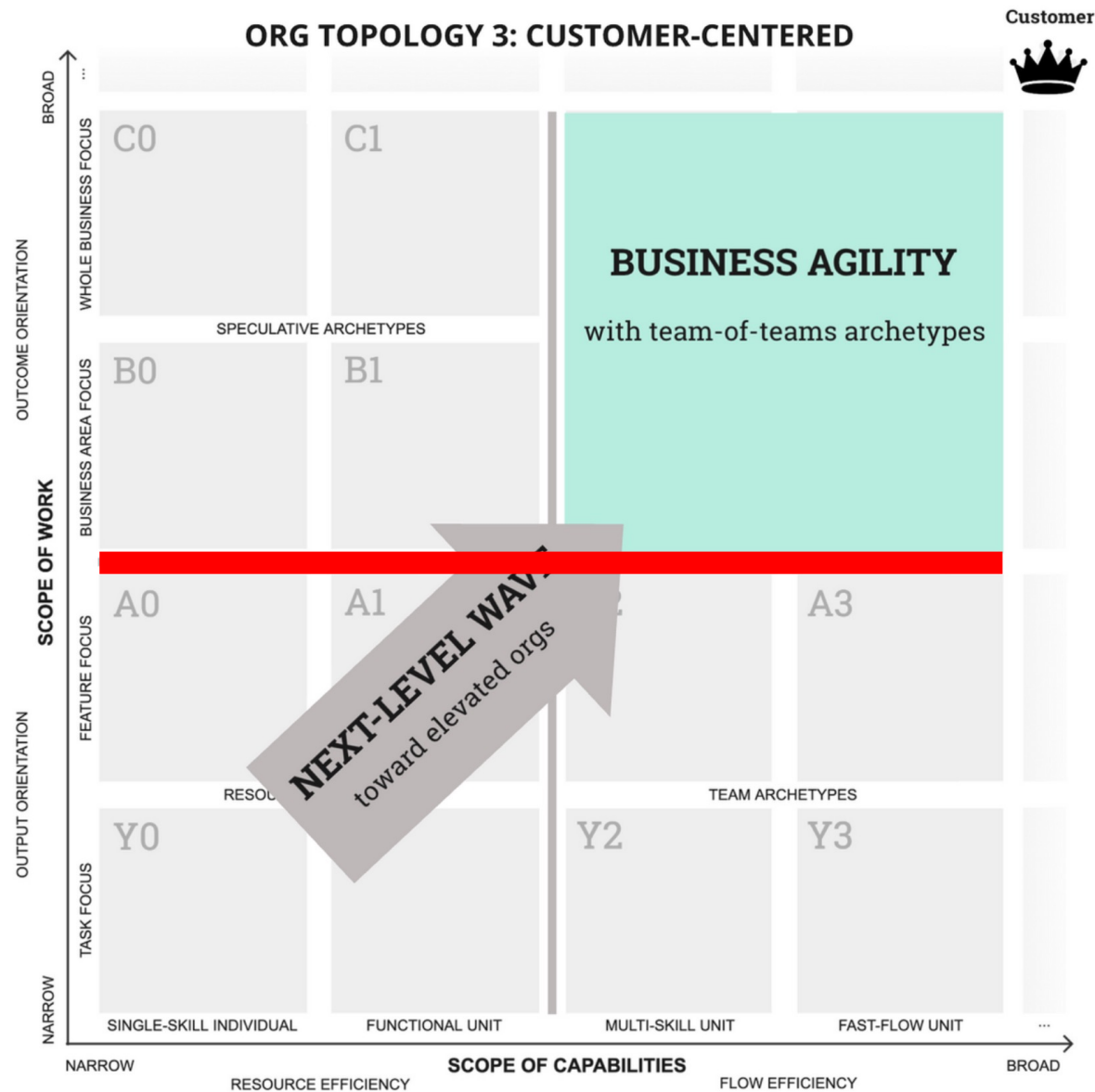
<https://itrevolution.com/articles/minimize-cognitive-load-of-teams/>





This is a misunderstanding.

Limit the size of software services/products to the cognitive load that the team can handle.



“Higher level archetypes”

- Outcome-oriented
- Flexible org design
- Team-of-teams working on high value

“Lower level archetypes”

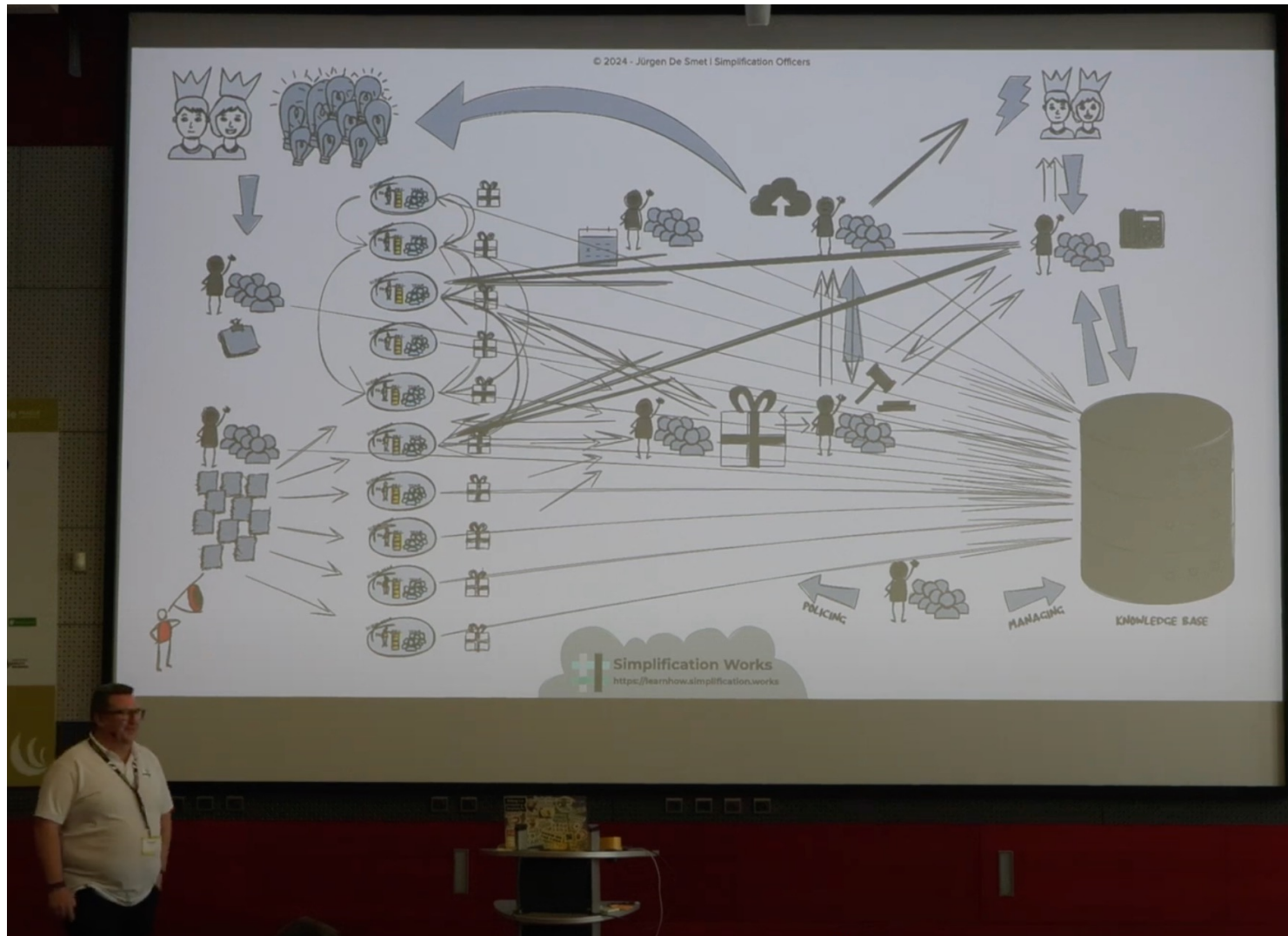
- Output-oriented
- Teams attached to architecture
- Reinforcement of Conway’s Law

We know now that CL:

1. deals with the short-term memory to store for processing
2. and the effort to learn and organize information

Doesn't necessarily increase with the size of the codebase to handle (unless you need to learn it by heart).

Doesn't increase with the amount of components to make changes in (unless the changes need to be made simultaneously).

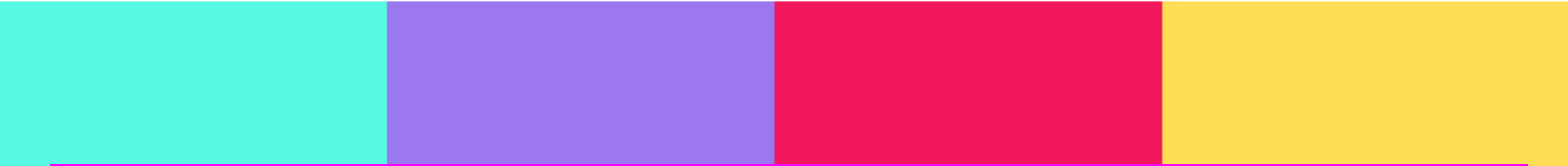


Teams with narrow ownership:

- have higher WIP
- need to track dependencies
- have many interruptions
- suffer context switching
- do not see the big picture
- rarely can work as a real team
- ...

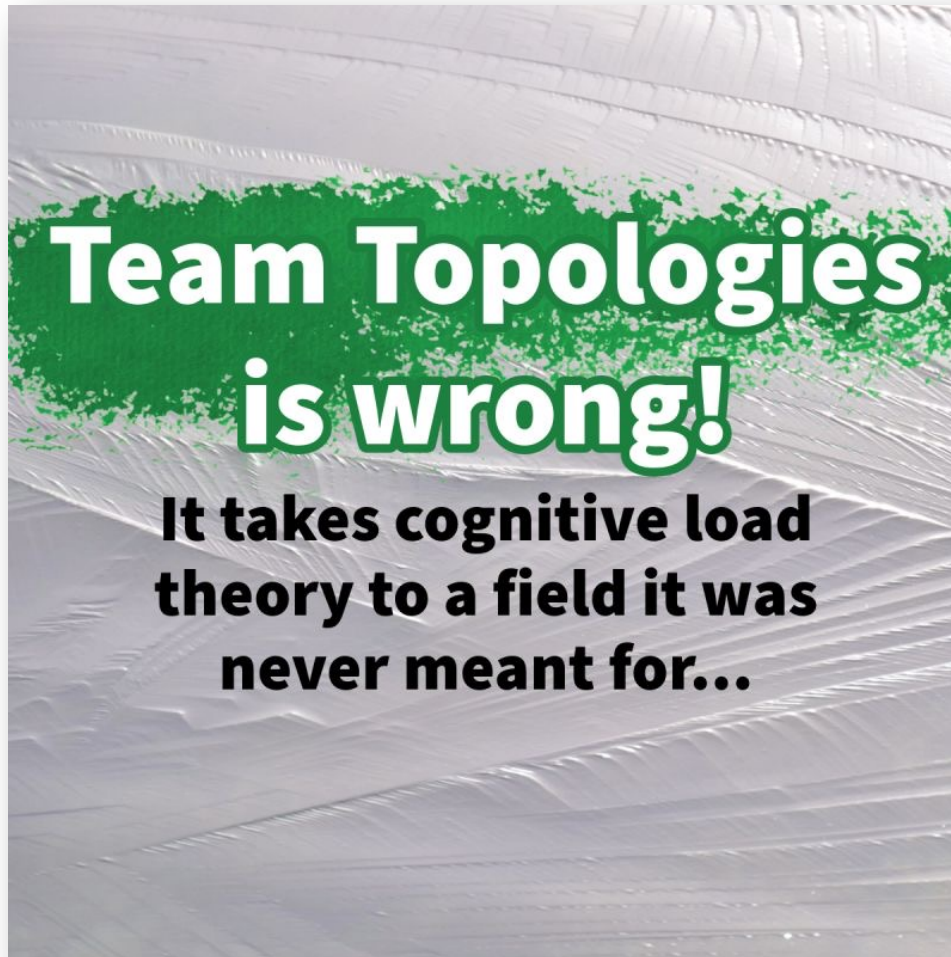
Jurgen de Smet at Agile Prague 2024

<https://www.youtube.com/watch?v=cS6ROMkoqCs>



This can also backfire and lead to the opposite effect

Limit the size of software services/products to the cognitive load that the team can handle.



Dennis Wagner (He/Him) • 1st

Full Stack Agile and Lean Transformation Expert

Book an appointment

3w •

...

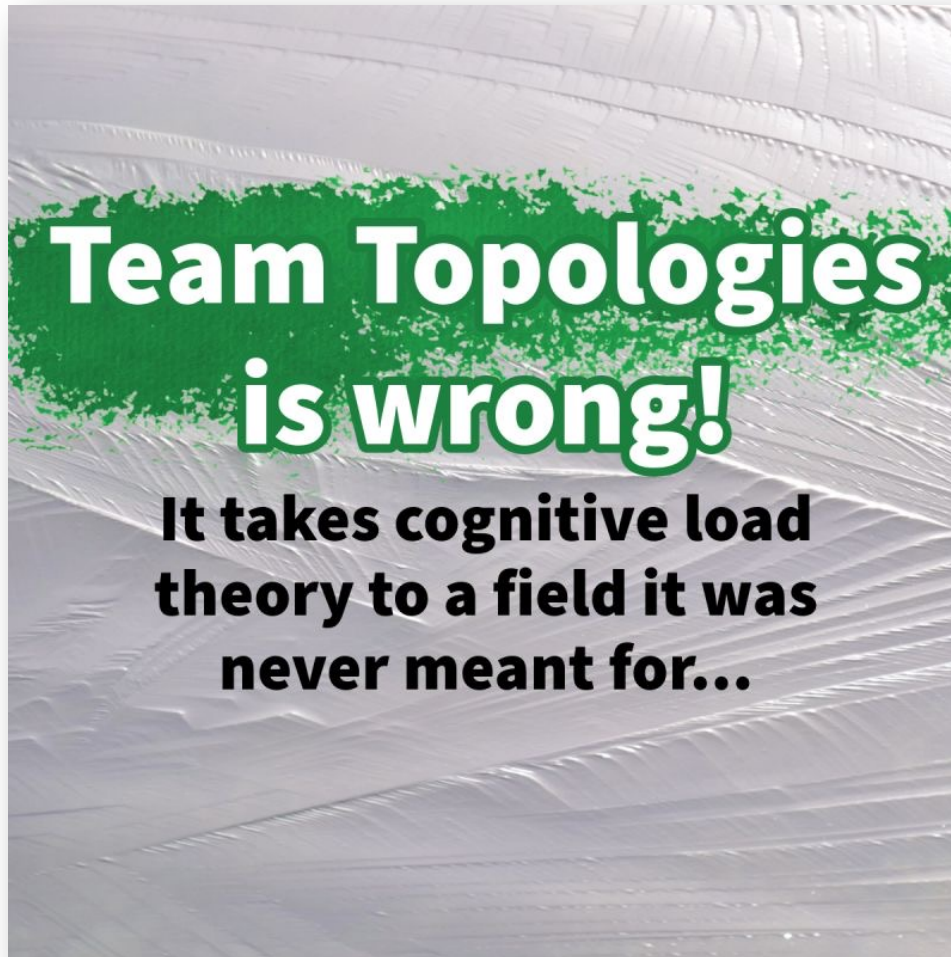
Debunking Cognitive Load Theory (CLT) in the Context of Team Work

Recently, I've engaged in discussions around Agile methodologies and the validity of certain long-held beliefs within these communities (like in this post by [Nicole Weidner](https://lnkd.in/epWcHccZ) <https://lnkd.in/epWcHccZ>). This led me to reflect deeply on another contentious topic: the application of Cognitive Load Theory (CLT) to team work, as proposed by the authors of Team Topologies. Here are (some of) my thoughts:

What is Cognitive Load Theory?

CLT originates from the field of educational psychology. It posits that learners have a limited cognitive capacity, which can be overwhelmed by too much information at once, hindering their ability to learn effectively. While this theory has its place in individual learning contexts, it faces considerable criticism and is not without its detractors, even in that field (e.g. <https://lnkd.in/ePCSCQAC>).

https://www.linkedin.com/posts/denniswagnercoach_devops-cognitiveloadtheory-teamwork-activity-7217450032990425088-0ujB



🚩 The Danger of Misapplying CLT:

The conclusions drawn by the authors of Team Topologies are not just wrong, they are dangerous! Misapplying CLT to team work undermines the essence of DevOps and Agile principles. It suggests a rigid, narrow approach to teamwork that will stifle innovation and flexibility!

As someone passionate about Agile methodologies, I urge the community to critically evaluate such theories before adopting them. Let's continue to explore and advocate for various approaches to team dynamics, including dynamic reteaming, to foster practical, evidence-based practices in our Agile and DevOps communities.

Feel free to share your thoughts and experiences on this topic. Let's foster a dialogue that promotes practical, evidence-based practices in our Agile and DevOps communities.

https://www.linkedin.com/posts/denniswagnercoach_devops-cognitiveloadtheory-teamwork-activity-7217450032990425088-0ujB

> This could be the end of this talk.

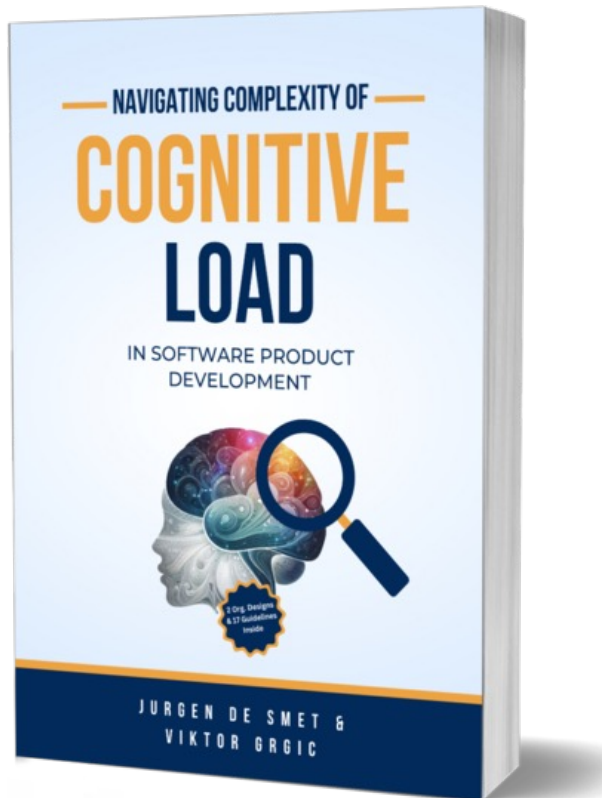


https://www.youtube.com/watch?v=3Vz_YTNLn6w



1:23,60

> This is actually a relevant topic!

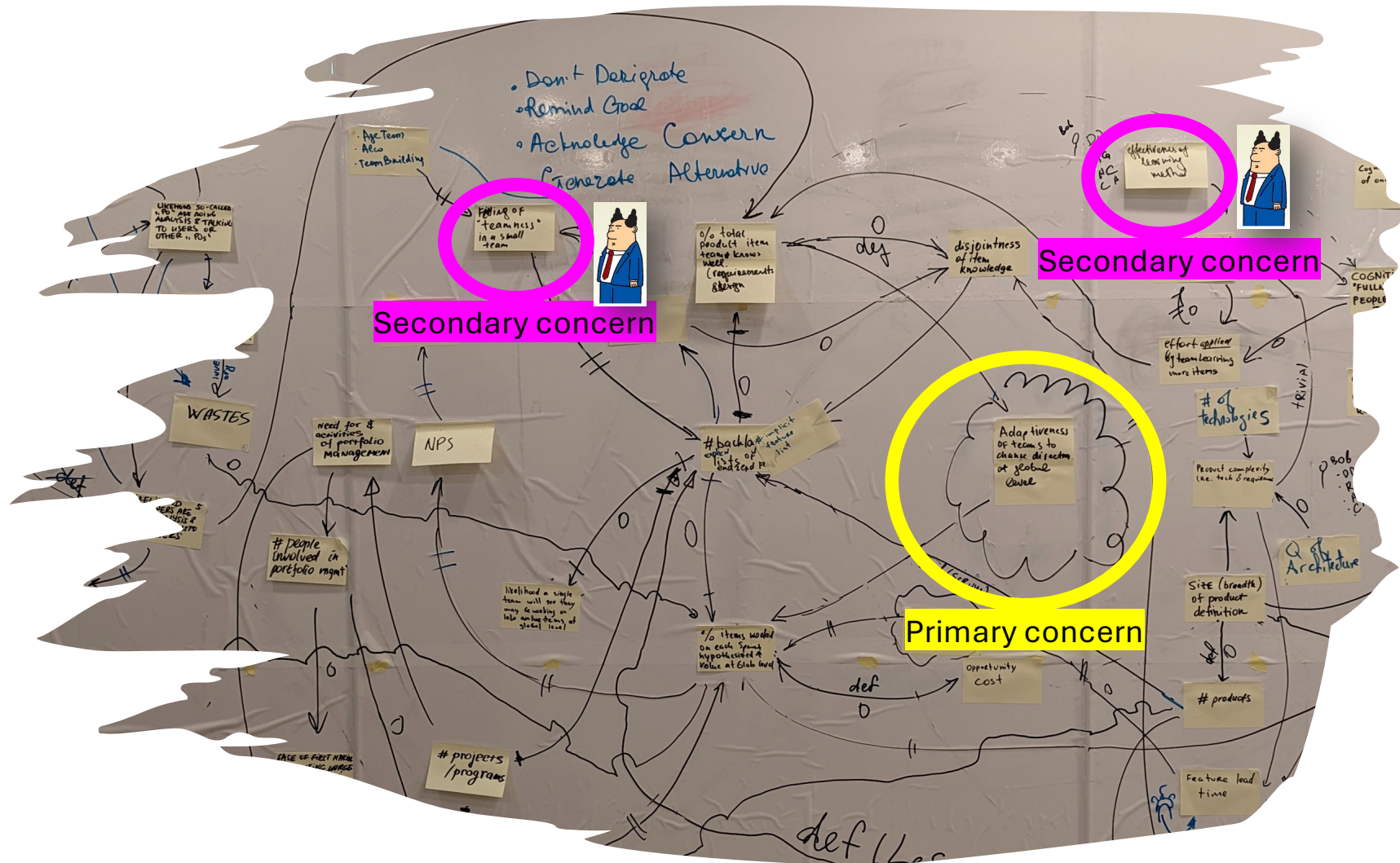


Foreword by Alexey Krivitsky (Org Topologies)

Have you ever tried collaborating with an overloaded person? In my experience, it is impossible to tell whether a person is overloaded or lacking professional attitude and skills. In other words, overloading makes us incapable of producing work of the expected quality.

The [American Institute of Stress](#) reports that workers' daily stress (measured in the US and Canada) increases by roughly 5% every year. Moreover, 83% of US workers suffer from work-related stress, with 25% saying their job is the number one stressor in their lives. Over time, on average, a stressed worker becomes 41% less productive and 33% less engaged.

learnhow.simplification.works/p/cognitive-load



From Craig's CLP class

Working with secondary concerns

- Don't denigrate or argue [expect good intent]
- Appreciate the stated concern [make people feel heard]
- Remind of the primary concern [CEO's messaging]
- Engage in finding ways to optimize for the secondary concerns *without* sub-optimizing the primary one(s) [avoid false dichotomies]



Our goal as org designers is to generate options for bringing the cognitive load down – without sub-optimizing the primary concern (adaptiveness, customer value, speed of value delivery...).

Imagine: you are **evil** managers



What can you do to maximize your workers' cognitive load?

OK. Un-evil yourselves now, please.



Most of these ideas can be reversed and drive CL down

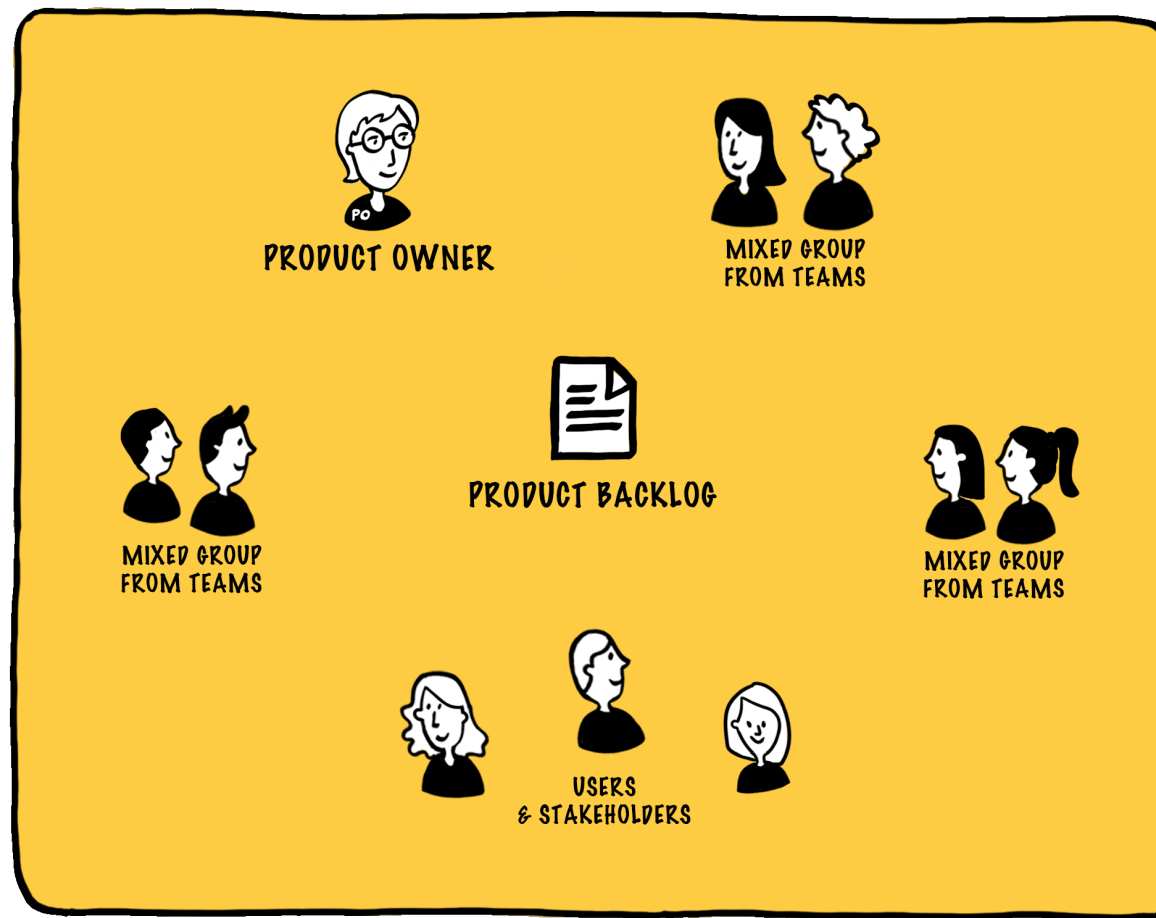
- TRY... AVOID ... Lack of whole product focus and proximity to the customer
- TRY... Connecting the teams to customers for empathy and requirement clarification from first-hand
- TRY... Testing automation
- TRY... AVOID ... Spreading team members across locations
- TRY... Collocated teams talking face-to-face
- TRY... AVOID ... Team coordinators and “fake POs”
- TRY... Teams coordinating
- TRY... TDD as taking small steps in solving complex problems
- TRY... AVOID ... Individuals owning “tickets”
- TRY... Teams sharing the work to improve focus, minimize WIP, speed up delivery, spread the load

Learning is an
intrinsic part of
product
development.

Multi-team PBR sessions at CollectAI ->



LESS MULTI-TEAM PRODUCT BACKLOG REFINEMENT



- Start by placing a cushioned layer on a stable, elevated platform. Gently position the tiny corpus onto this surface, ensuring they are resting comfortably on their dorsal side. Release the adhesive fasteners on the existing protective covering that encases the lower half of their form. Carefully remove the soiled sheath, lifting it away to reveal the affected area.
- With the soiled covering set aside, use pre-moistened, soft wipes to meticulously cleanse the exposed region, paying particular attention to any folds and crevices. Ensure all residues are thoroughly removed, leaving the surface clean and dry.
- Once the cleansing is complete, retrieve a fresh, absorbent wrap designed for the lower extremities. Unfold and prepare this new covering, positioning it beneath the tiny individual. Secure the covering by aligning and fastening the adhesive tabs snugly but comfortably around the midsection.
- Dispose of the used materials in a designated container for waste items. Finally, ensure the tiny individual is comfortable and return them to their usual environment, where they can continue their activities with a refreshed and clean lower covering.



HTTP Status Codes

Registration Procedure(s)

IETF Review

Reference

[RFC9110, Section 16.2.1]

Note

1xx: Informational – Request received, continuing process
2xx: Success – The action was successfully received, understood, and accepted
3xx: Redirection – Further action must be taken in order to complete the request
4xx: Client Error – The request contains bad syntax or cannot be fulfilled
5xx: Server Error – The server failed to fulfill an apparently valid request

Available Formats



csv

Value	Description	Reference
100	Continue	[RFC9110, Section 15.2.1]
101	Switching Protocols	[RFC9110, Section 15.2.2]
102	Processing	[RFC2518]
103	Early Hints	[RFC8297]
104-199	Unassigned	
200	OK	[RFC9110, Section 15.3.1]
201	Created	[RFC9110, Section 15.3.2]
202	Accepted	[RFC9110, Section 15.3.3]
203	Non-Authoritative Information	[RFC9110, Section 15.3.4]
204	No Content	[RFC9110, Section 15.3.5]
205	Reset Content	[RFC9110, Section 15.3.6]
206	Partial Content	[RFC9110, Section 15.3.7]
207	Multi-Status	[RFC4918]
208	Already Reported	[RFC5842]
209-225	Unassigned	
226	IM Used	[RFC3229]
227-299	Unassigned	
300	Multiple Choices	[RFC9110, Section 15.4.1]
301	Moved Permanently	[RFC9110, Section 15.4.2]
302	Found	[RFC9110, Section 15.4.3]
303	See Other	[RFC9110, Section 15.4.4]
304	Not Modified	[RFC9110, Section 15.4.5]
305	Use Proxy	[RFC9110, Section 15.4.6]
306	(Unused)	[RFC9110, Section 15.4.7]
307	Temporary Redirect	[RFC9110, Section 15.4.8]
308	Permanent Redirect	[RFC9110, Section 15.4.9]

309-399	Unassigned	
400	Bad Request	[RFC9110, Section 15.5.1]
401	Unauthorized	[RFC9110, Section 15.5.2]
402	Payment Required	[RFC9110, Section 15.5.3]
403	Forbidden	[RFC9110, Section 15.5.4]
404	Not Found	[RFC9110, Section 15.5.5]
405	Method Not Allowed	[RFC9110, Section 15.5.6]
406	Not Acceptable	[RFC9110, Section 15.5.7]
407	Proxy Authentication Required	[RFC9110, Section 15.5.8]
408	Request Timeout	[RFC9110, Section 15.5.9]
409	Conflict	[RFC9110, Section 15.5.10]
410	Gone	[RFC9110, Section 15.5.11]
411	Length Required	[RFC9110, Section 15.5.12]
412	Precondition Failed	[RFC9110, Section 15.5.13]
413	Content Too Large	[RFC9110, Section 15.5.14]
414	URI Too Long	[RFC9110, Section 15.5.15]
415	Unsupported Media Type	[RFC9110, Section 15.5.16]
416	Range Not Satisfiable	[RFC9110, Section 15.5.17]
417	Expectation Failed	[RFC9110, Section 15.5.18]
418	(Unused)	[RFC9110, Section 15.5.19]
419-420	Unassigned	
421	Misdirected Request	[RFC9110, Section 15.5.20]
422	Unprocessable Content	[RFC9110, Section 15.5.21]
423	Locked	[RFC4918]
424	Failed Dependency	[RFC4918]
425	Too Early	[RFC8470]
426	Upgrade Required	[RFC9110, Section 15.5.22]
427	Unassigned	
428	Precondition Required	[RFC6585]
429	Too Many Requests	[RFC6585]
430	Unassigned	
431	Request Header Fields Too Large	[RFC6585]
432-450	Unassigned	
451	Unavailable For Legal Reasons	[RFC7725]
452-499	Unassigned	
500	Internal Server Error	[RFC9110, Section 15.6.1]
501	Not Implemented	[RFC9110, Section 15.6.2]
502	Bad Gateway	[RFC9110, Section 15.6.3]
503	Service Unavailable	[RFC9110, Section 15.6.4]
504	Gateway Timeout	[RFC9110, Section 15.6.5]



Steve Losh
@stevelosh

HTTP status ranges in a nutshell:

1xx: hold on
2xx: here you go
3xx: go away
4xx: you fucked up
5xx: I fucked up

HTTP Status Codes		
Registration Procedure(s) IETF Review		
Reference [RFC9110, Section 16.2.1]		
Note 1xx: Informational – Request received, continuing process 2xx: Success – The action was successfully received, understood, and accepted 3xx: Redirection – Further action must be taken in order to complete the request 4xx: Client Error – The request contains bad syntax or cannot be fulfilled 5xx: Server Error – The server failed to fulfill an apparently valid request		
Available Formats		
Value	Description	Reference
100	Continue	[RFC9110, Section 15.2.1]
101	Switching Protocols	[RFC9110, Section 15.2.2]
102	Processing	[RFC2518]
103	Early Hints	[RFC8297]
104-199	Unassigned	[RFC9110, Section 15.3.1]
200	OK	[RFC9110, Section 15.3.2]
201	Created	[RFC9110, Section 15.3.3]
202	Accepted	[RFC9110, Section 15.3.4]
203	Non-Authoritative Information	[RFC9110, Section 15.3.5]
204	No Content	[RFC9110, Section 15.3.6]
205	Reset Content	[RFC9110, Section 15.3.7]
206	Partial Content	[RFC4918]
207	Multi-Status	[RFC5842]
208	Already Reported	[RFC3229]
209-225	Unassigned	[RFC9110, Section 15.4.1]
226	IM Used	[RFC9110, Section 15.4.2]
227-299	Unassigned	[RFC9110, Section 15.4.3]
300	Multiple Choices	[RFC9110, Section 15.4.4]
301	Moved Permanently	[RFC9110, Section 15.4.5]
302	Found	[RFC9110, Section 15.4.6]
303	See Other	[RFC9110, Section 15.4.7]
304	Not Modified	[RFC9110, Section 15.4.8]
305	Use Proxy	[RFC9110, Section 15.4.9]
306	(Unused)	[RFC9110, Section 15.4.10]
307	Temporary Redirect	[RFC9110, Section 15.4.11]
308	Permanent Redirect	[RFC9110, Section 15.4.12]
309-399	Unassigned	[RFC9110, Section 15.5.1]
400	Bad Request	[RFC9110, Section 15.5.2]
401	Unauthorized	[RFC9110, Section 15.5.3]
402	Payment Required	[RFC9110, Section 15.5.4]
403	Forbidden	[RFC9110, Section 15.5.5]
404	Not Found	[RFC9110, Section 15.5.6]
405	Method Not Allowed	[RFC9110, Section 15.5.7]
406	Not Acceptable	[RFC9110, Section 15.5.8]
407	Proxy Authentication Required	[RFC9110, Section 15.5.9]
408	Request Timeout	[RFC9110, Section 15.5.10]
409	Conflict	[RFC9110, Section 15.5.11]
410	Gone	[RFC9110, Section 15.5.12]
411	Length Required	[RFC9110, Section 15.5.13]
412	Precondition Failed	[RFC9110, Section 15.5.14]
413	Content Too Large	[RFC9110, Section 15.5.15]
414	URI Too Long	[RFC9110, Section 15.5.16]
415	Unsupported Media Type	[RFC9110, Section 15.5.17]
416	Range Not Satisfiable	[RFC9110, Section 15.5.18]
417	Expectation Failed	[RFC9110, Section 15.5.19]
418	(Unused)	[RFC9110, Section 15.5.20]
419-420	Unassigned	[RFC9110, Section 15.5.21]
421	Misdirected Request	[RFC4918]
422	Unprocessable Content	[RFC4918]
423	Locked	[RFC4918]
424	Failed Dependency	[RFC4918]
425	Too Early	[RFC6585]
426	Upgrade Required	[RFC6585]
427	Unassigned	[RFC6585]
428	Precondition Required	[RFC6585]
429	Too Many Requests	[RFC6585]
430	Unassigned	[RFC6585]
431	Request Header Fields Too Large	[RFC7725]
432-450	Unassigned	[RFC9110, Section 15.6.1]
451	Unavailable For Legal Reasons	[RFC9110, Section 15.6.2]
452-499	Unassigned	[RFC9110, Section 15.6.3]
500	Internal Server Error	[RFC9110, Section 15.6.4]
501	Not Implemented	[RFC9110, Section 15.6.5]
502	Bad Gateway	[RFC9110, Section 15.6.6]
503	Service Unavailable	[RFC9110, Section 15.6.7]
504	Gateway Timeout	[RFC9110, Section 15.6.8]

Strategies to Optimize Cognitive Load for Better Learning and Memory

1. Simplify Complex Information: Break down complex information into smaller, manageable parts to reduce intrinsic load.

Use customer language

“Take a bite”

Start with explaining how the product works now

Strategies to Optimize Cognitive Load for Better Learning and Memory

1. Simplify Complex Information: Break down complex information into smaller, manageable parts to reduce intrinsic load.

Use customer language

“Take a bite”

Start with explaining how the product works now

2. Improve Instructional Design: Use clear, concise instructions and relevant materials to minimize extraneous load.

Start with the why

Connect teams with customers

Strategies to Optimize Cognitive Load for Better Learning and Memory

1. Simplify Complex Information: Break down complex information into smaller, manageable parts to reduce intrinsic load.

Use customer language

“Take a bite”

Start with explaining how the product works now

2. Improve Instructional Design: Use clear, concise instructions and relevant materials to minimize extraneous load.

Start with the why

Connect teams with customers

3. Encourage Active Processing: Engage learners in activities that promote deep processing (germane load) without overwhelming them.

Test understanding with SbE

Work on freshly refined items next sprint

Keep refining the remaining part next PBRs

Strategies to Optimize Cognitive Load for Better Learning and Memory

1. Simplify Complex Information: Break down complex information into smaller, manageable parts to reduce intrinsic load.

Use customer language

“Take a bite”

Start with explaining how the product works now

2. Improve Instructional Design: Use clear, concise instructions and relevant materials to minimize extraneous load.

Start with the why

Connect teams with customers

3. Encourage Active Processing: Engage learners in activities that promote deep processing (germane load) without overwhelming them.

Test understanding with SbE

Work on freshly refined items next sprint

Keep refining the remaining part next PBRs

4. Use Visual Aids and Examples: Visual aids, analogies, and practical examples can help manage intrinsic and germane loads.

User whiteboards (not Jira)

Strategies to Optimize Cognitive Load for Better Learning and Memory

1. Simplify Complex Information: Break down complex information into smaller, manageable parts to reduce intrinsic load.

Use customer language

“Take a bite”

Start with explaining how the product works now

2. Improve Instructional Design: Use clear, concise instructions and relevant materials to minimize extraneous load.

Start with the why

Connect teams with customers

3. Encourage Active Processing: Engage learners in activities that promote deep processing (germane load) without overwhelming them.

Test understanding with SbE

Work on freshly refined items next sprint

Keep refining the remaining part next PBRs

4. Use Visual Aids and Examples: Visual aids, analogies, and practical examples can help manage intrinsic and germane loads.

User whiteboards (not Jira)

5. Allow Time for Reflection: Give learners time to process and reflect on information to support schema development.

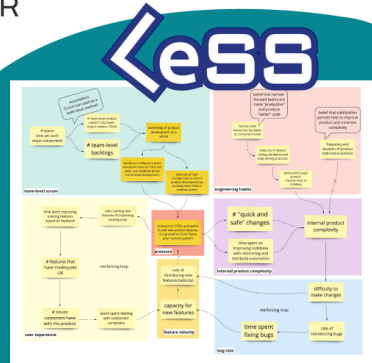
Run PBR one per sprint

LESS ADOPTION AT POSTER POS INC.

IN TIMES OF COVID-19 AND WAR

PART 1 ORG BEFORE CHANGE

BY ALEXEY KRIVITSKY

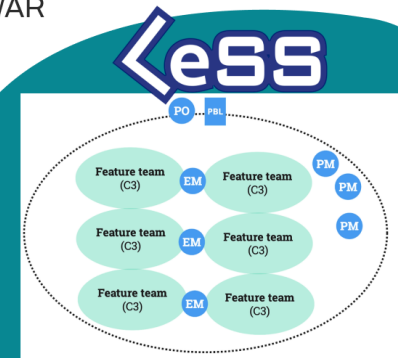


LESS ADOPTION AT POSTER POS INC.

IN TIMES OF COVID-19 AND WAR

PART 2 DESIGNING NEW ORG

BY ALEXEY KRIVITSKY



LESS ADOPTION AT POSTER POS INC.

IN TIMES OF COVID-19 AND WAR

PART 3 FLIPPING THE ORG

BY ALEXEY KRIVITSKY

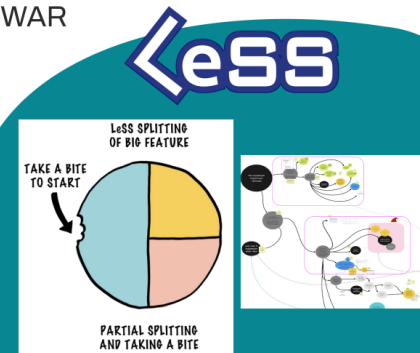


LESS ADOPTION AT POSTER POS INC.

IN TIMES OF COVID-19 AND WAR

PART 4 TWENTY LESS SPRINTS

BY ALEXEY KRIVITSKY



miro | LeSS Conference 2023 Talk

37:42 / 46:20

4

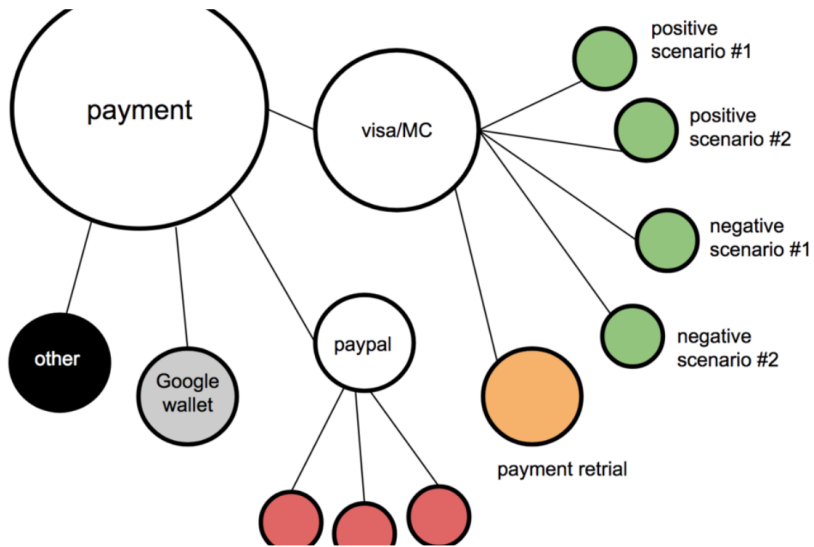
Alexey Krivitsky

LeSS adoption at Poster:
ouches, gotchas and yooahoo

LeSS
conference
2023

<https://www.youtube.com/watch?v=rqVTsHvyMWY>

“PBR” topics start at 33:00



What if I tell you requirements are like bubbles?



Alexey Krivitsky

CTO | Org Design Consultant | Product & Engineering Coach
| Org Topologies co-creator | Lego4Scrum book author |...



April 19, 2018

'Epic' is too good of a name for a piece of uncertainty that could fail you so badly!

Comments

170 · 27 comments



Add a comment...



Most relevant ▾



Pamela Velásquez Sierra · 2nd

10mo ...

Sr Business Analyst at Endava | Sr Product Ma...

graphs and colors? I love it!

Like | Reply



Dan Rawsthorne, Ph...

2y (edited) ...

· 2nd

Chief Scientist and Senior Agility Instructor at ...

Looks like a Use Case Map from the UML. I always liked them... back in the '90s we referred to this technique as "the ever-unfolding story"... I taught a lot of classes on it 20 years ago... also involves commonality/variability analysis. ...more

Like · 1 | Reply



Alexey Krivitsky Author

4y ...

CTO | Org Design Consultant | Product & Engi...

My fresh video on how to keep both: bigger picture and good granularity in a Product Backlog. <https://www.youtube.com/watch?v=dsfhLME-9jI>

<https://www.linkedin.com/pulse/what-i-tell-yourequirements-like-bubbles-alexey-krivitsky>

> How many techniques did we count so far?



More ideas?

Continuous Integration (CI) as a Practice
Communicate in Code
Just Talk

Use modelling techniques for the big picture (“Current Architecture Workshops”)

Cross-functional work within a team
Avoid separating discovery and delivery (a.k.a. “Continuous Discovery Habits”)
Embed business analysis into the teams

Fast feedback
Retrospective
Stable iterations

Pairing / mobbing
Simplify, automate and standardize

Scrum masters to remove systemic impediments and interruptions
Minimize general organizational complexity

Use GenAI

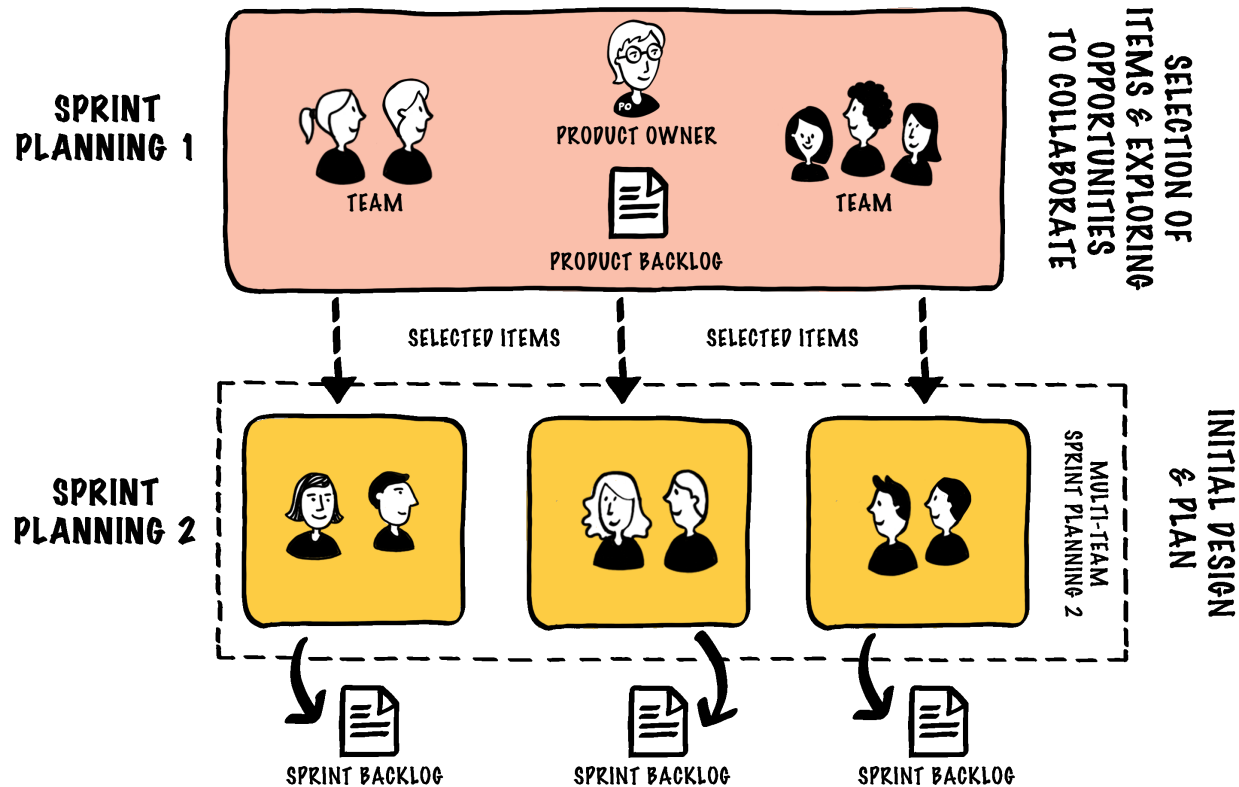
...

> How many techniques did we count so far?

> Wait... but there's more!

$$\text{ROI in Learning} = \frac{\text{Value of Knowledge Application (VOKA)}}{\text{Cost of Knowledge Acquisition (COKA)}}$$

LESS SPRINT PLANNING



During SP 1, the Teams are agreeing to pull work from the Product Backlog:

- they follow priorities (increase VOKA)
- and try to benefit from pre-existing knowledge (decrease COKA)

29 TECHNIQUES FOR REDUCING THE COGNITIVE LOAD

that you won't hear
them talking about



Alexey
Krivitsky

Org Consultant
De/Scaling Expert
Product & Engineering Coach

who is standing on the shoulders of the giants

