# EvoCrawl: Exploring Web Application Code and State using Evolutionary Search

**Xiangyu Guo**, Akshay Kawlay, Eric Liu, David Lie

## Background

EvoCrawl is a black-box crawler detecting vulnerabilities in client-server web applications by interacting with the browser on the client-side.

- Black-box: the crawler has no access to the source code
- Why Black-box: More generalized since web applications have been developed in many different programming languages

## How do we improve Code Coverage

Performance of a crawler largely depends on how much code it can cover.

How can we improve code coverage for a crawler with no access to source code?

- **Number of crawled pages**: Heavily Influenced by the **Application States**

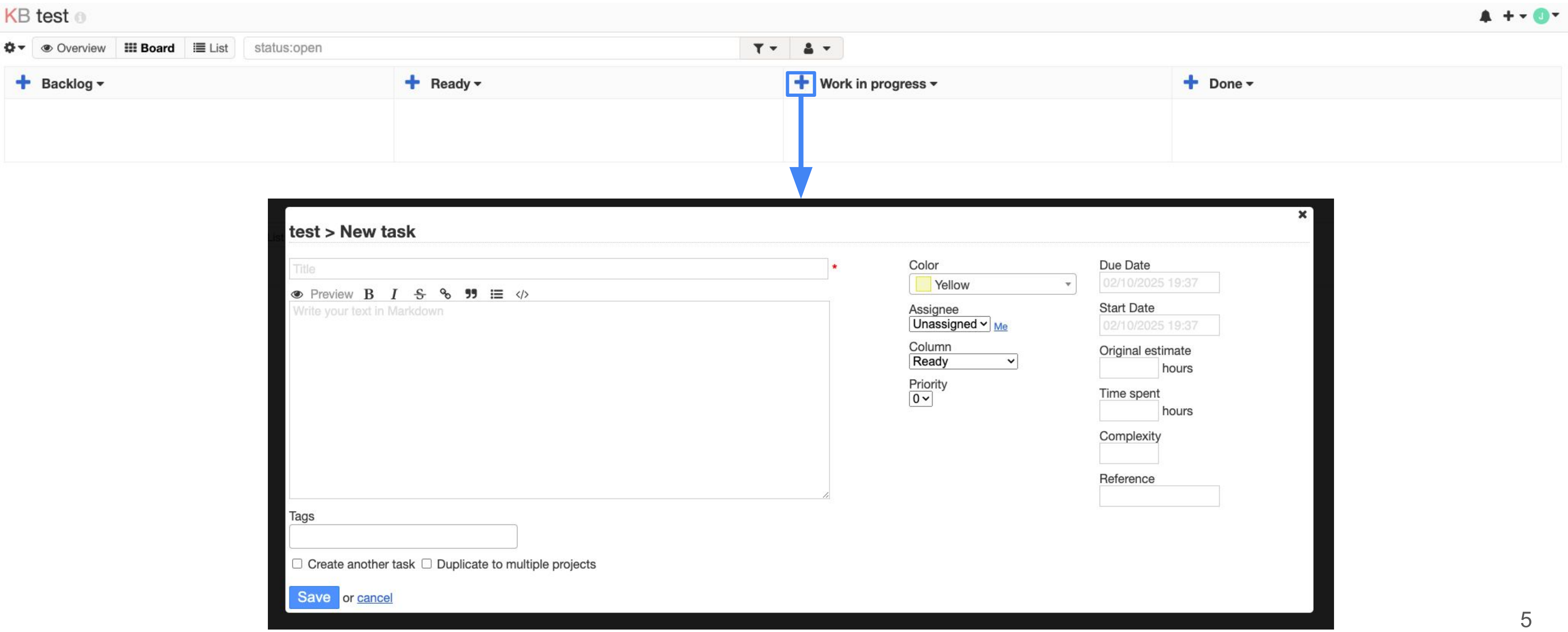
    **Certain Pages can only be crawled when the application are in certain states**

## It is still a difficult task to fully explore the states

How does a black-box scanner change the data stored on the server:

- Submit inputs
- Most common method: Submitting HTML forms

# Submitting HTML forms can be hard



status:open

Color: Yellow

Assignee: Unassigned

Column: Ready

Priority: 0

Due Date: 02/10/2025 19:37

Start Date: 02/10/2025 19:37

Original estimate: hours

Time spent: hours

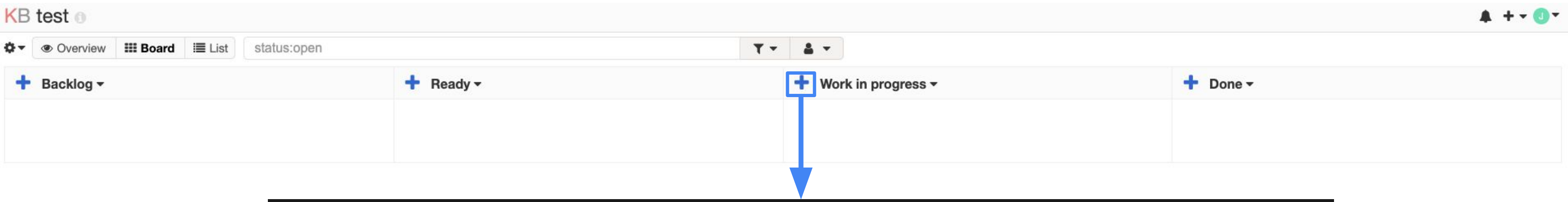
Complexity

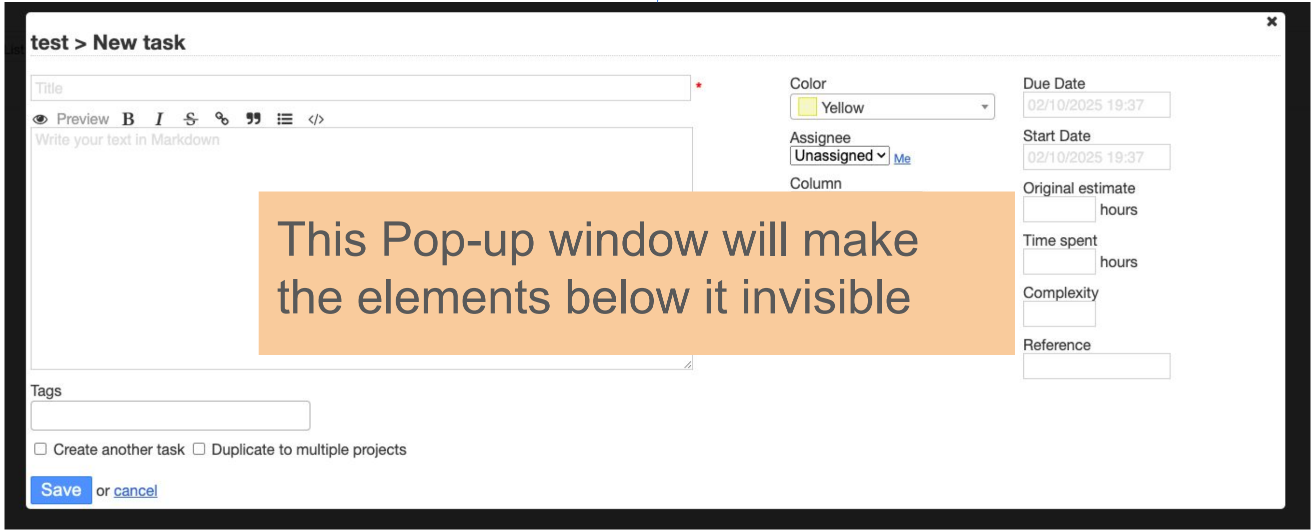
Reference

Tags

☐ Create another task ☐ Duplicate to multiple projects

# Submitting HTML forms can be hard





test > New task

Title *

Preview

Write your text in Markdown

Color: Yellow

Assignee: Unassigned Me

Column

Due Date: 02/10/2025 19:37

Start Date: 02/10/2025 19:37

Original estimate hours

Time spent hours

Complexity

Reference

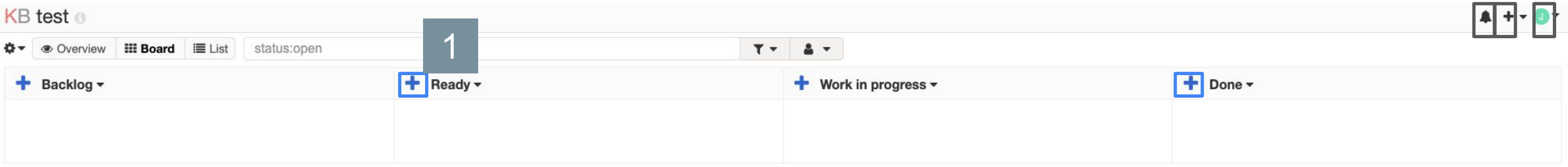
Tags

☐ Create another task ☐ Duplicate to multiple projects

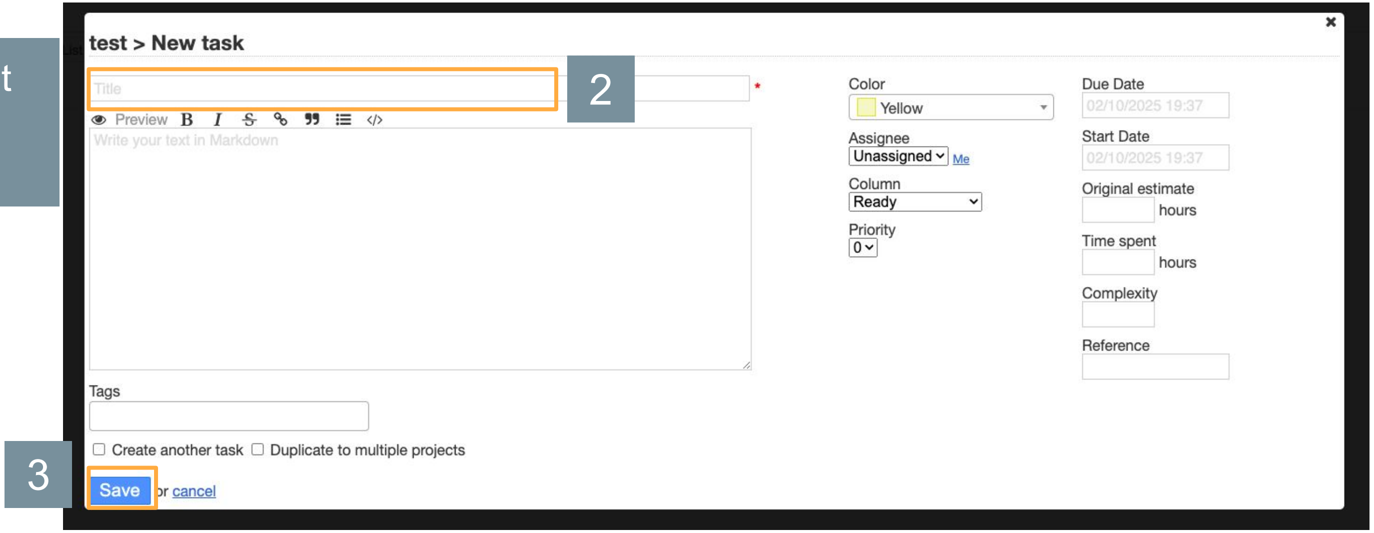
Save or cancel

This Pop-up window will make the elements below it invisible

# Submitting HTML forms is not trivial

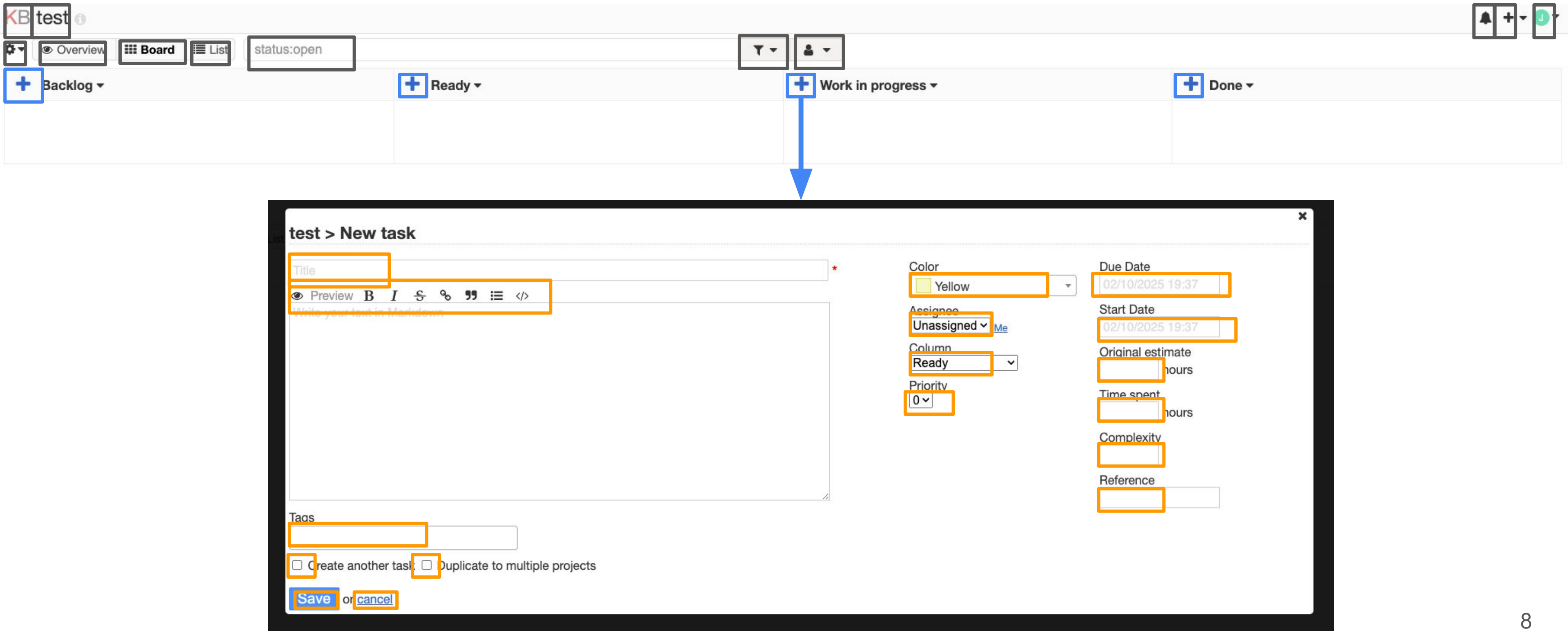


The Simplest Sequence Interaction

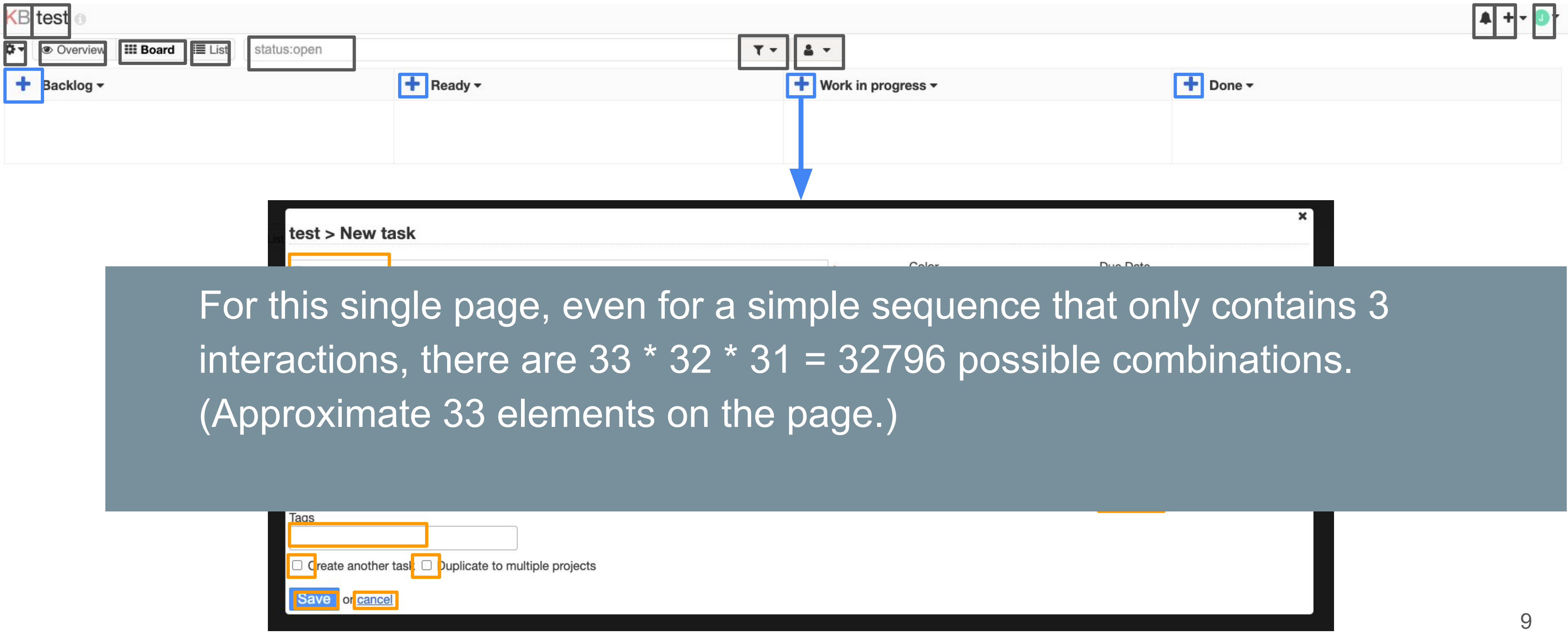


| | |
|---|---|
| Color | Yellow |
| Assignee | Unassigned |
| Column | Ready |
| Priority | 0 |
| Due Date | 02/10/2025 19:37 |
| Start Date | 02/10/2025 19:37 |
| Original estimate | hours |
| Time spent | hours |
| Complexity | |
| Reference | |

# Submitting HTML forms is not trivial

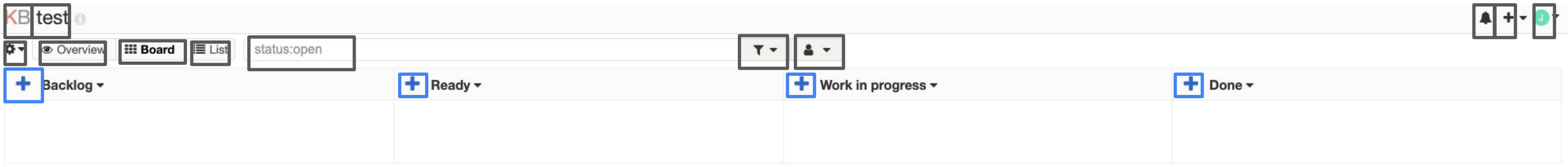


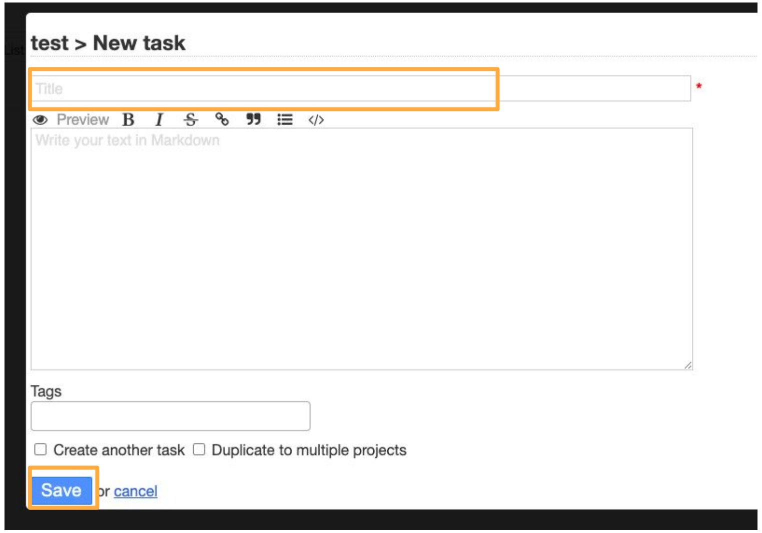
## Submitting HTML forms is not trivial



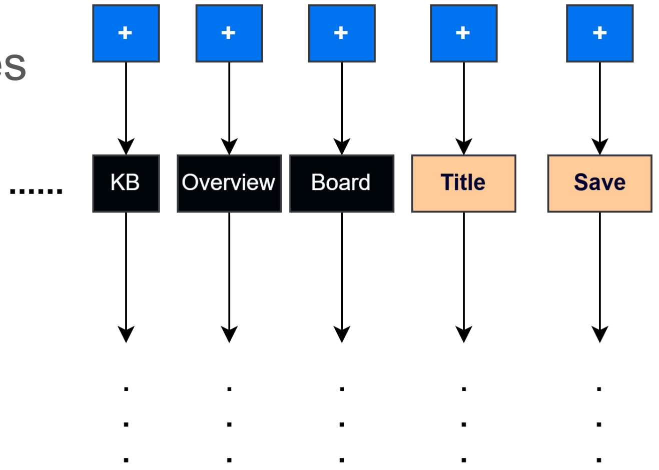
For this single page, even for a simple sequence that only contains 3 interactions, there are 33 * 32 * 31 = 32796 possible combinations. (Approximate 33 elements on the page.)

## Submitting HTML forms is not trivial - How to find the right sequence?

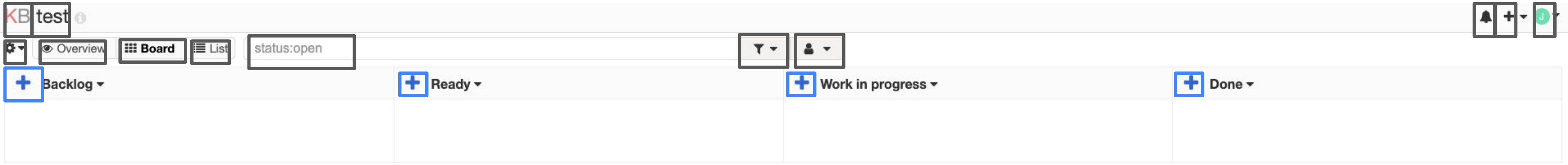


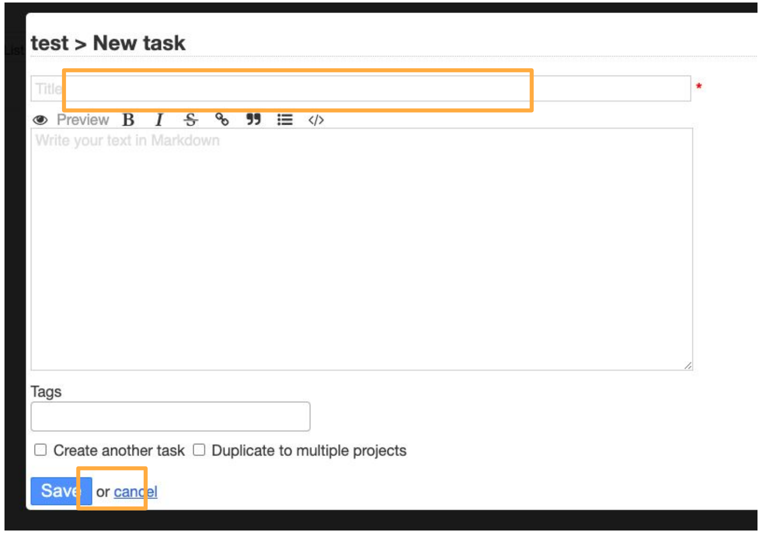


Possible Sequences



## Submitting HTML forms is not trivial - How to find the right sequence?




test > New task

Title

Preview

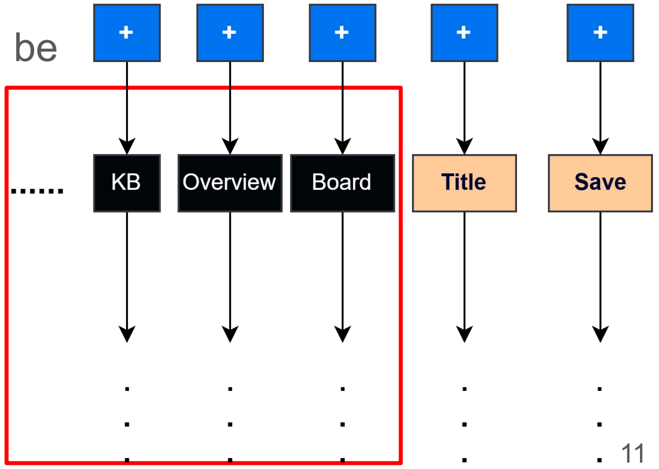
Write your text in Markdown

Tags

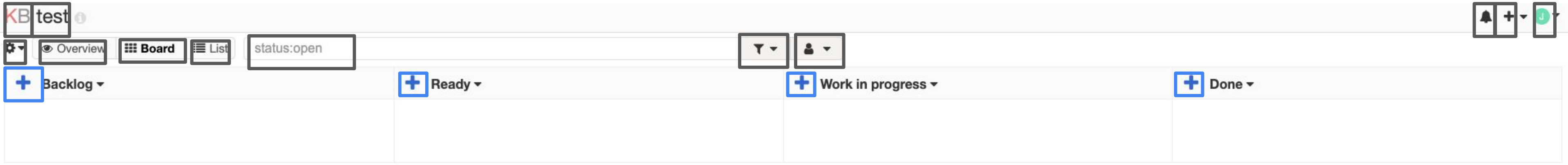
☐ Create another task ☐ Duplicate to multiple projects

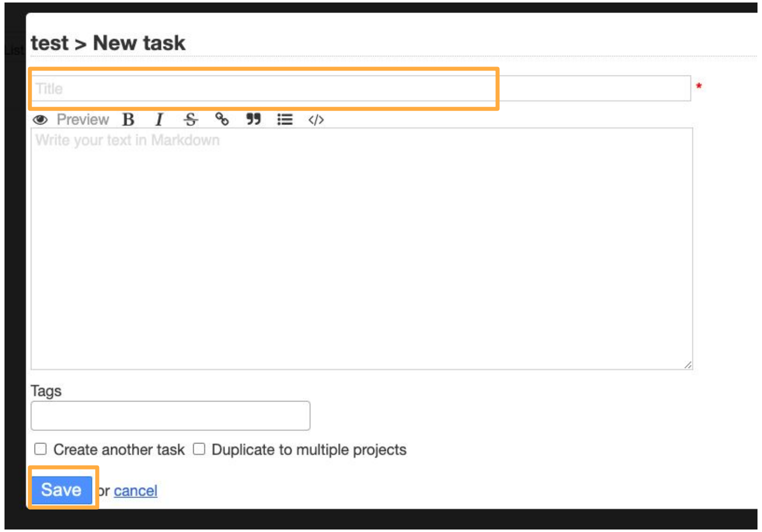
Save or cancel

Avoid sequences with bad orders:
Elements in black box will be **blocked by the pop-up window**

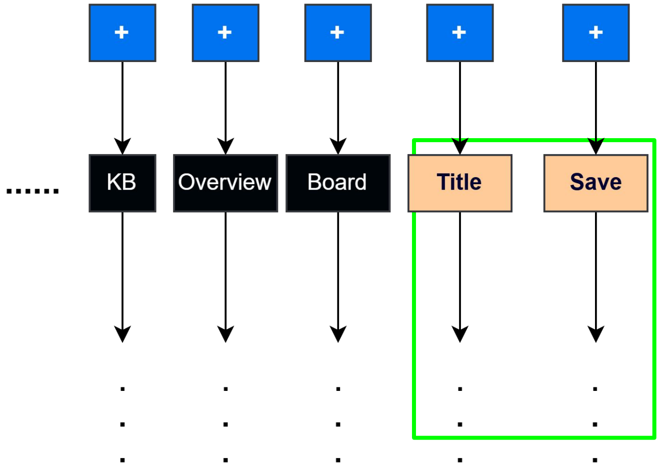


## Submitting HTML forms is not trivial - How to find the right sequence?

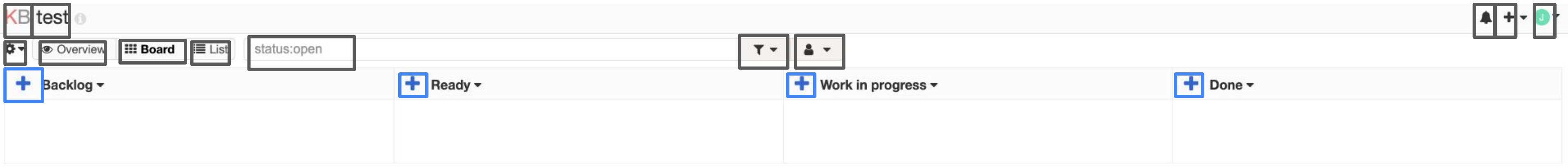


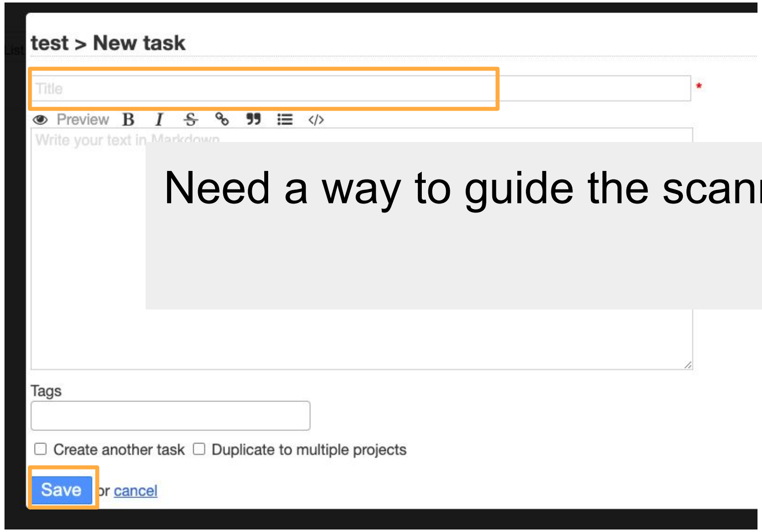

test > New task

Explore sequences with good orders



## Submitting HTML forms is not trivial - How to find the right sequence?




test > New task

Title

Preview B I

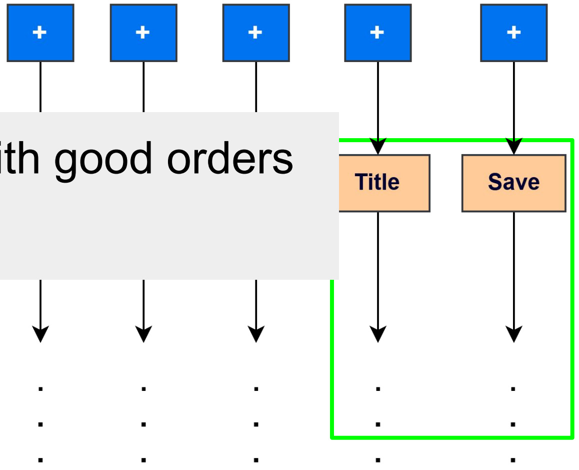
Write your text in Markdown

Tags

☐ Create another task ☐ Duplicate to multiple projects

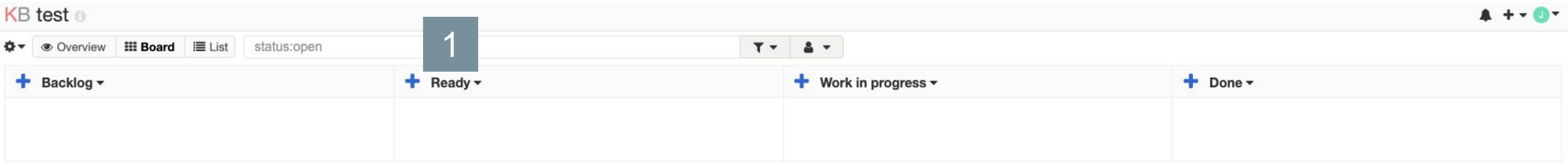
Save or cancel

Explore sequences with good orders

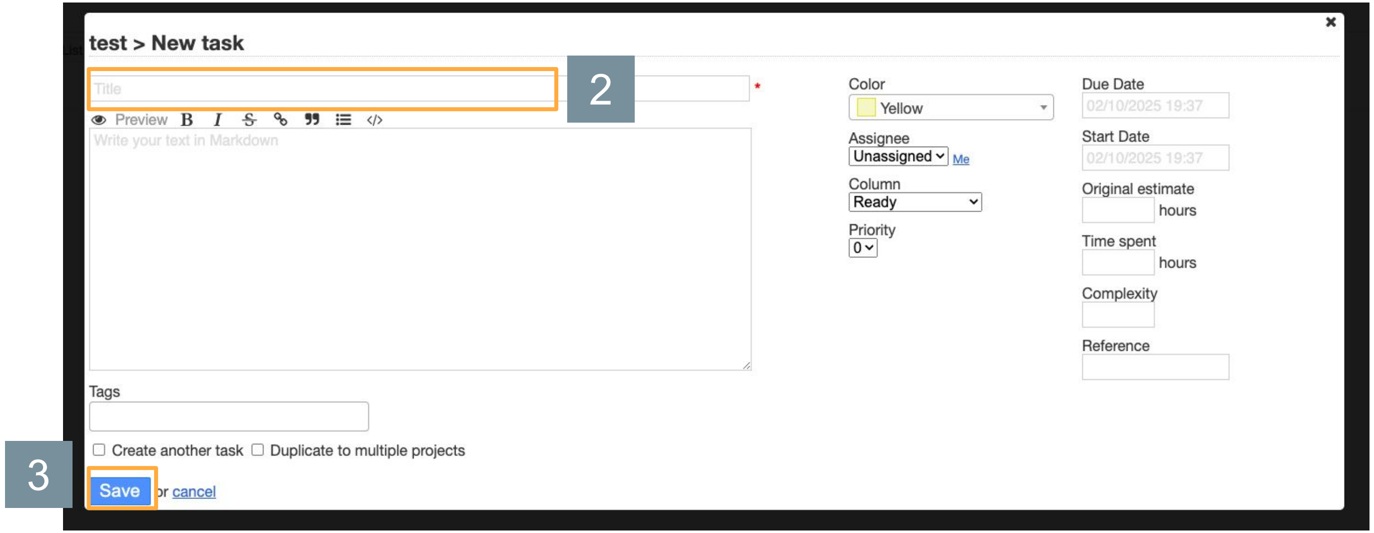


Need a way to guide the scanner to explore sequences with good orders

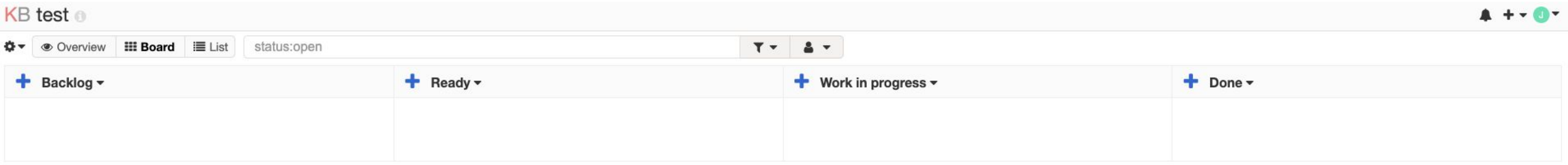
# Is finding one sequence enough?



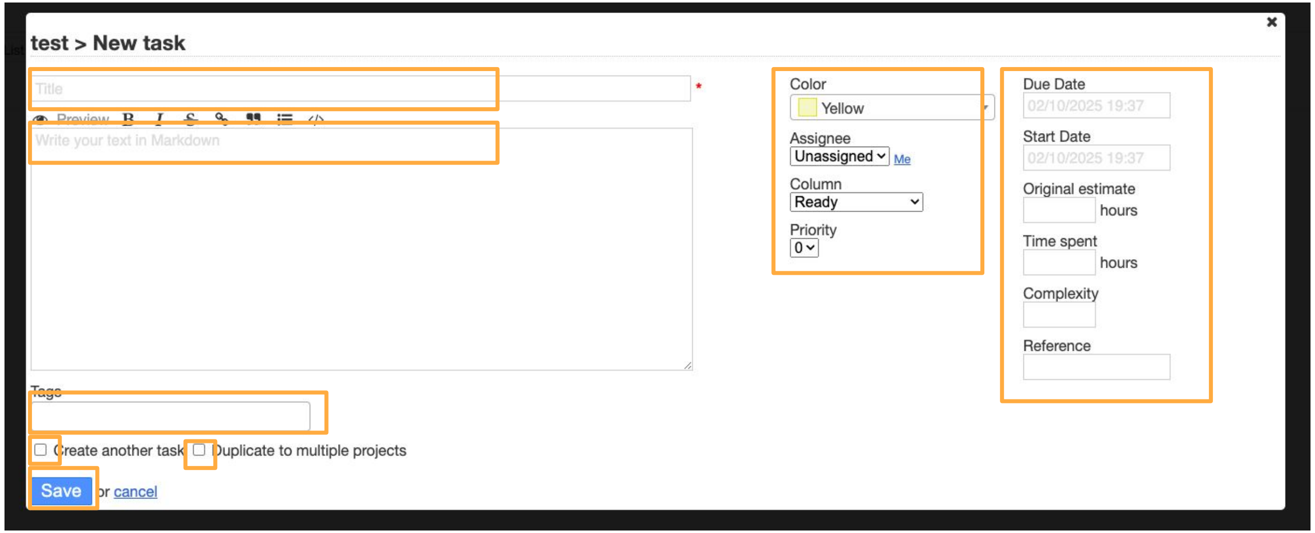
Should we stop at here?



# Is finding one sequence enough?



How about other elements inside the form?

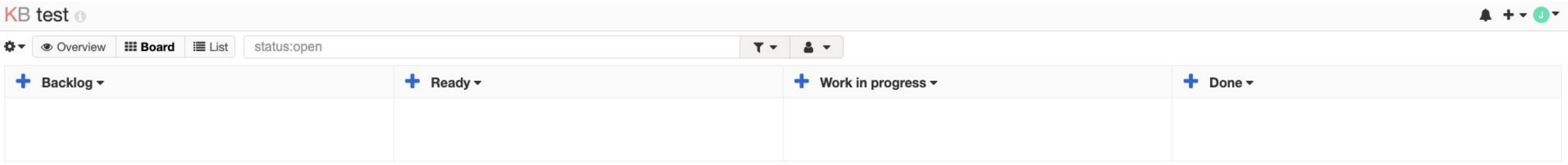

test > New task

| | |
|---|---|
| Title | |
| Write your text in Markdown | |
| Color | Yellow |
| Assignee | Unassigned |
| Column | Ready |
| Priority | 0 |
| Due Date | 02/10/2025 19:37 |
| Start Date | 02/10/2025 19:37 |
| Original estimate | hours |
| Time spent | hours |
| Complexity | |
| Reference | |
| Tags | |

☐ Create another task ☐ Duplicate to multiple projects

Save or cancel

# Is finding one sequence enough?

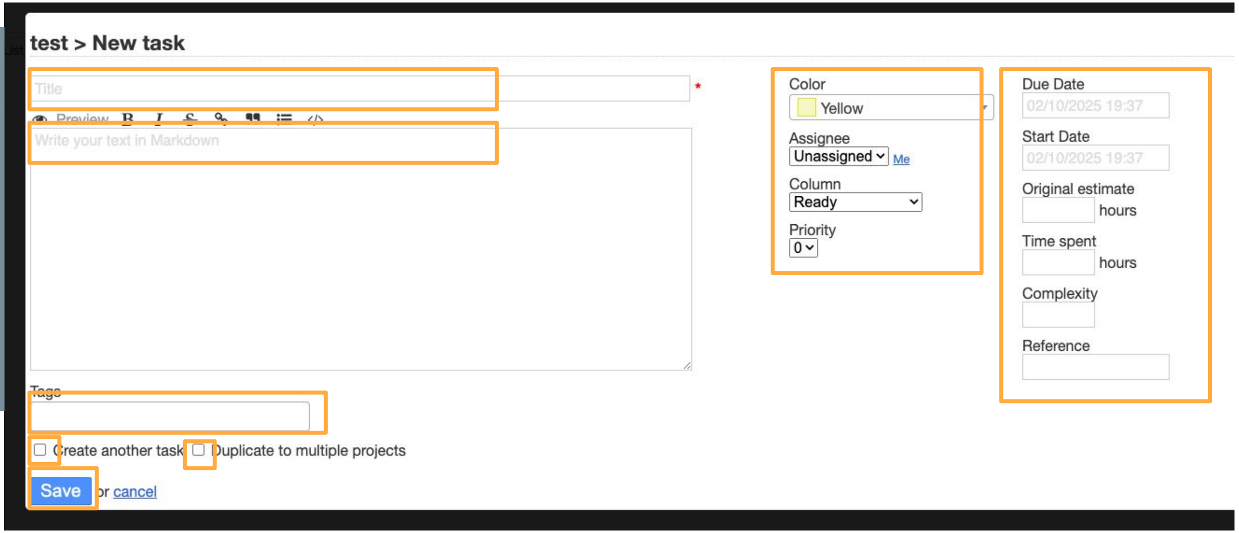


KB test

Overview | Board | List | status:open

Backlog | Ready | Work in progress | Done

- Form submission with different combination of inputs can trigger different codes on the server
- They can be injecting points for XSS payload



test > New task

Title *

Preview B I S

Write your text in Markdown

Tags

Create another task

Duplicate to multiple projects

Save or cancel

Color: Yellow

Assignee: Unassigned Me

Column: Ready

Priority: 0

Due Date: 02/10/2025 19:37

Start Date: 02/10/2025 19:37

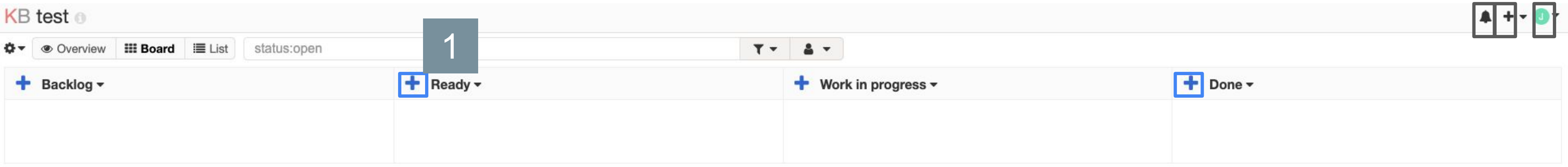
Original estimate hours

Time spent hours

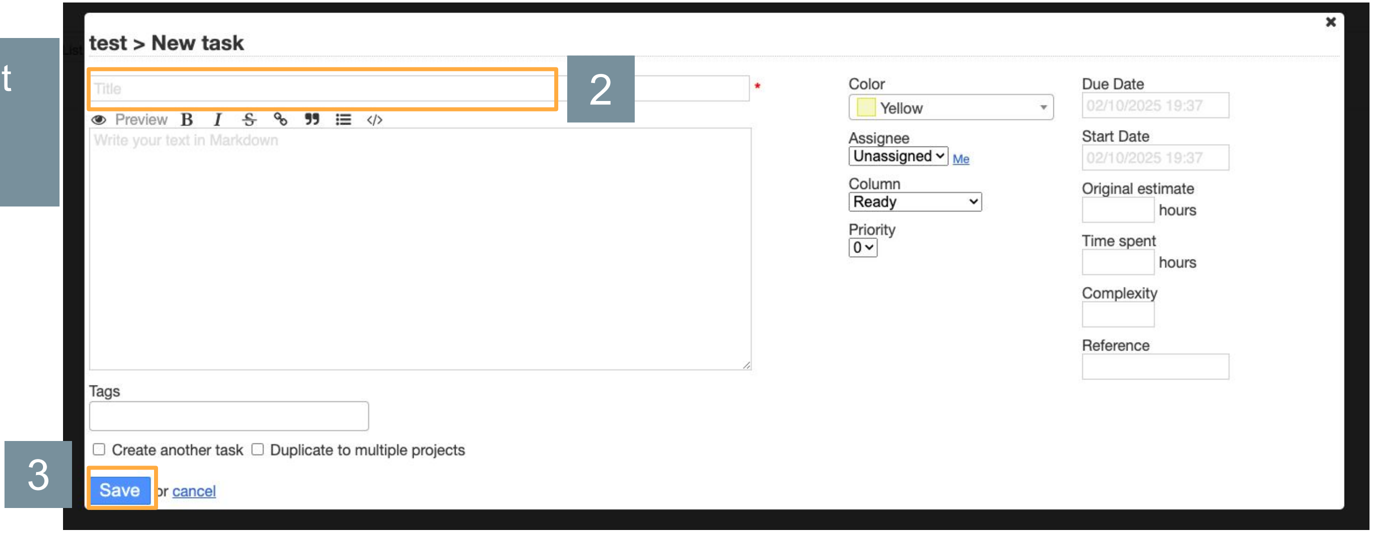
Complexity

Reference

# Submitting HTML forms is not trivial



The Simplest Sequence Interaction

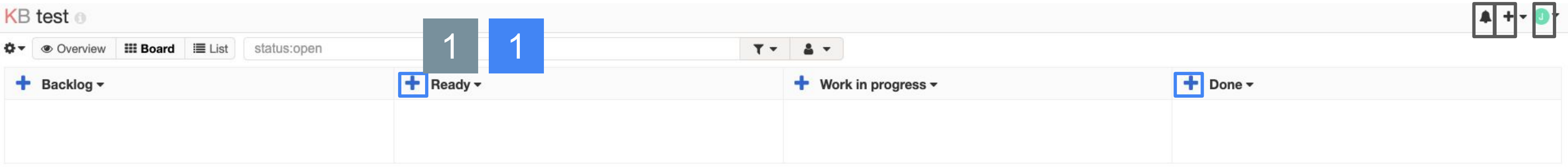


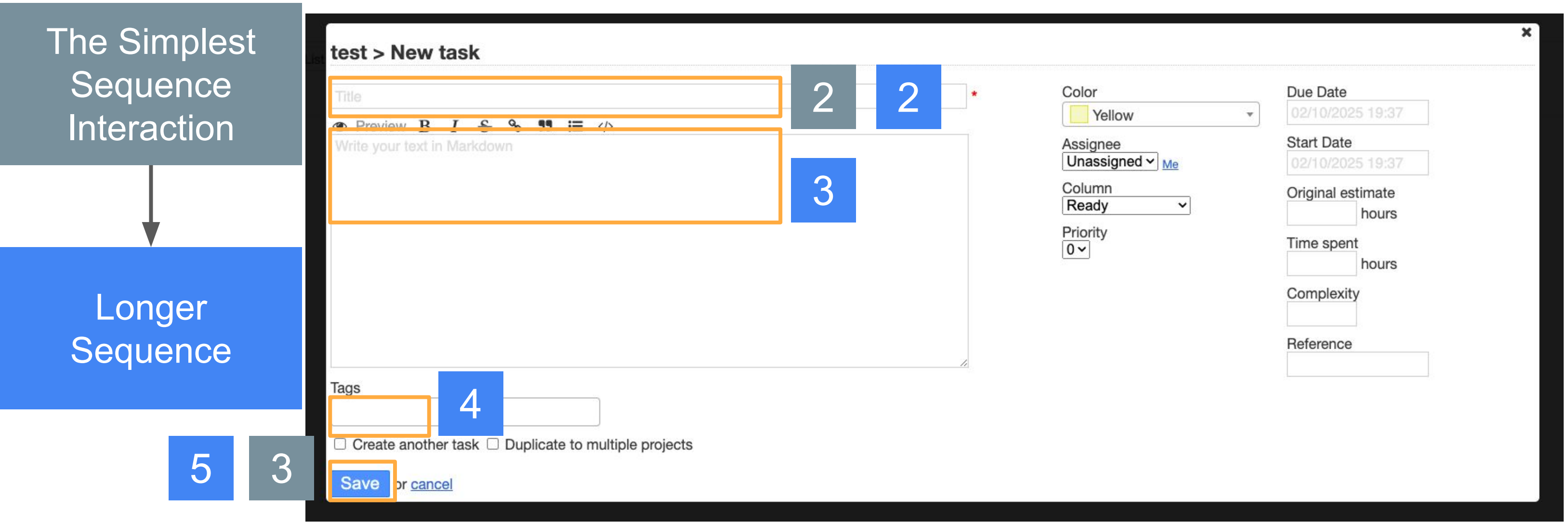
test > New task

| | |
|---|---|
| Title | * |
| Write your text in Markdown | |
| Tags | |
| Color | Yellow |
| Assignee | Unassigned (Me) |
| Column | Ready |
| Priority | 0 |
| Due Date | 02/10/2025 19:37 |
| Start Date | 02/10/2025 19:37 |
| Original estimate | hours |
| Time spent | hours |
| Complexity | |
| Reference | |

☐ Create another task ☐ Duplicate to multiple projects

# Submitting HTML forms is not trivial





test > New task

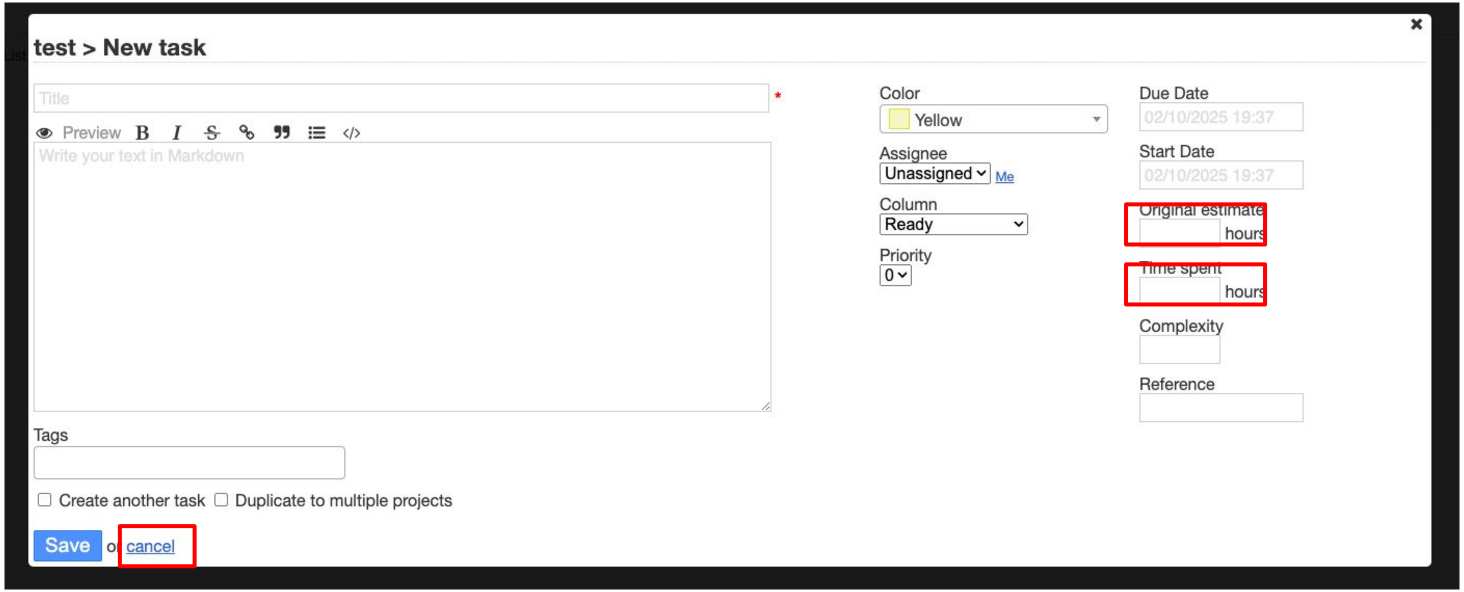
| | |
|---|---|
| Title | |
| Write your text in Markdown | |
| Color | Yellow |
| Assignee | Unassigned |
| Column | Ready |
| Priority | 0 |
| Due Date | 02/10/2025 19:37 |
| Start Date | 02/10/2025 19:37 |
| Original estimate | hours |
| Time spent | hours |
| Complexity | |
| Reference | |
| Tags | |

☐ Create another task ☐ Duplicate to multiple projects

# Submitting HTML forms is not trivial

Avoid certain elements: such us “cancel button” or elements that require values cannot be inferred by scanners

Partially Preserver the order: Make the “Save” button to be at the end, etc.



test > New task

Title *

Preview

Write your text in Markdown

Tags

☐ Create another task ☐ Duplicate to multiple projects

Save or cancel

| | |
|---|---|
| Color | Yellow |
| Assignee | Unassigned Me |
| Column | Ready |
| Priority | 0 |
| Due Date | 02/10/2025 19:37 |
| Start Date | 02/10/2025 19:37 |
| Original estimate | hours |
| Time spent | hours |
| Complexity | |
| Reference | |

## EvoCrawl is

- A way to effectively generate sequences that preserve good orders
- A way to generate new sequences from the sequences with good orders

## EvoCrawl is

- A way to effectively generate sequences that preserve good order
- A way to generate new sequences from the sequences with good order

Evolutionary Search and Dependency Tracking

# Diversified Evolutionary Search

**Fitness Function:** Identify Sequences With good order

Reward:

| | | |
|---|---|---|
| **Inject inputs** | **Fill in input fields** | **Trigger JavaScript Events** |

Punish:

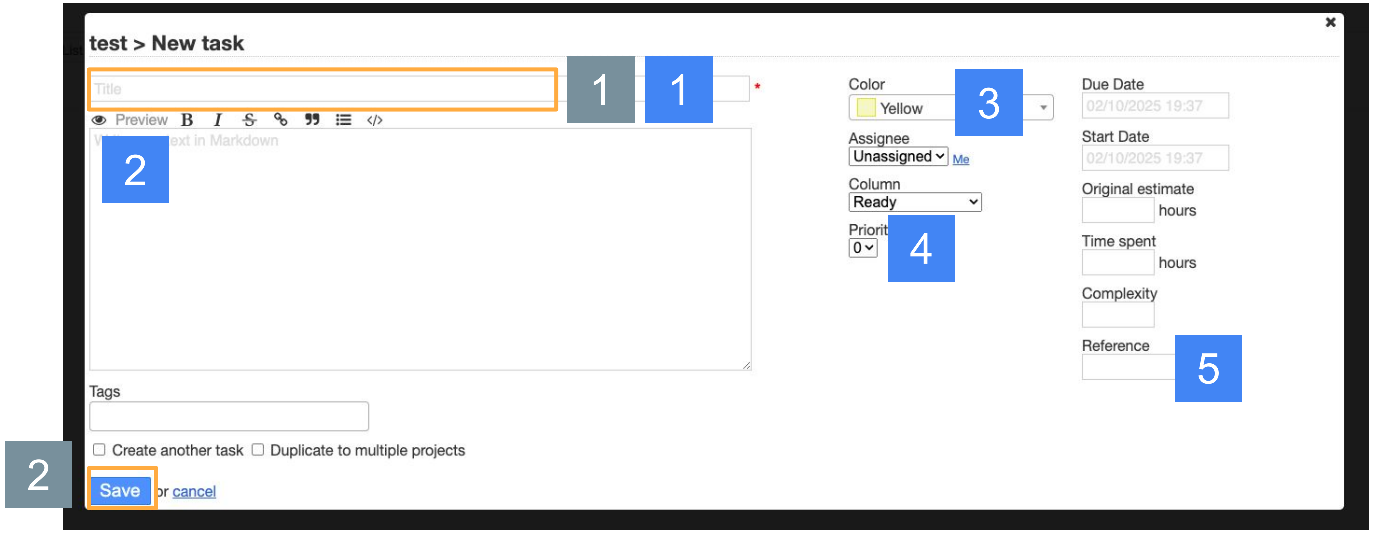
**Interact with blocked elements**

Aims to explore diversified states of the application -> Need Diversified Sequences

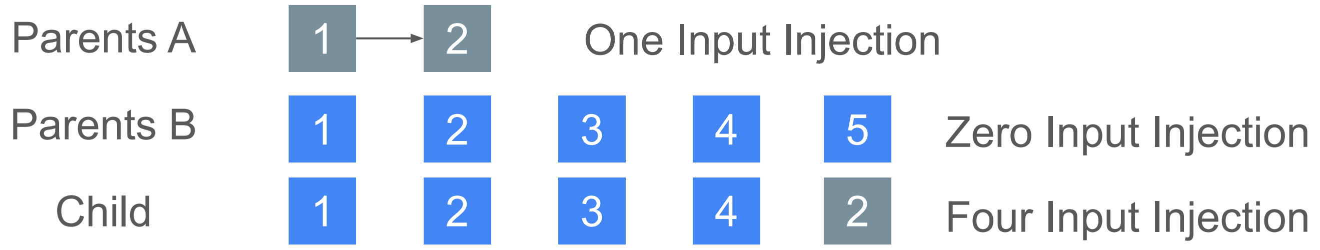
# Diversified Evolutionary Search

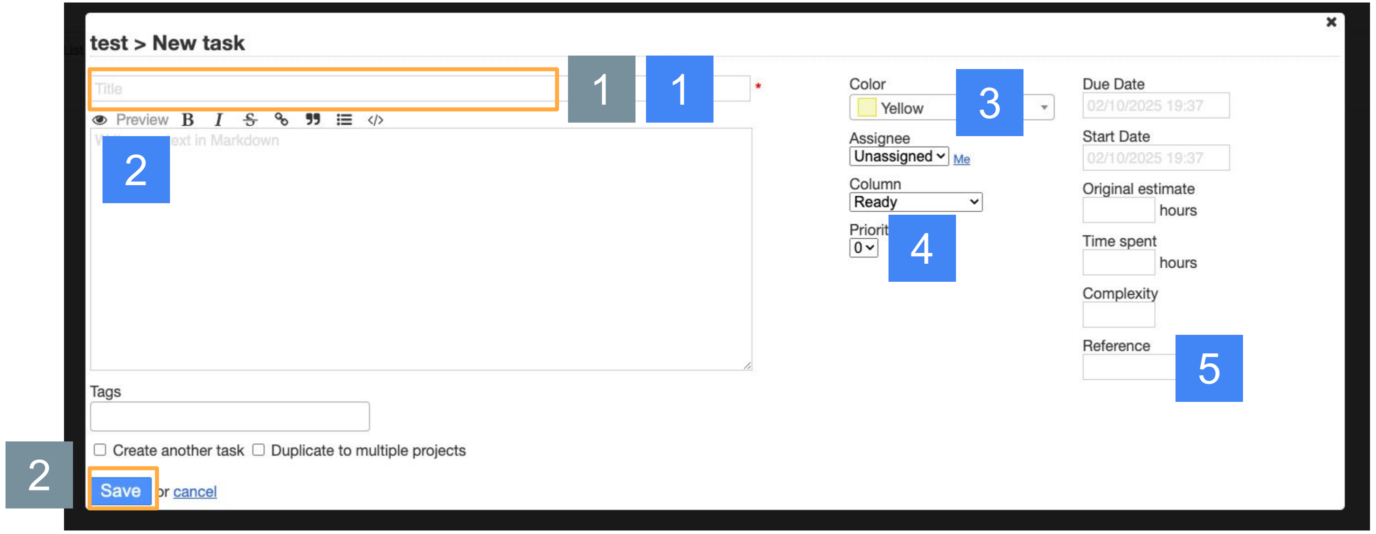
**Crossover:** Generate new sequences based on the Sequences with good order

- Concatenate parts of two sequences together: introducing diversity into sequences while partially maintaining the order



# Diversified Evolutionary Search





test > New task

Title

Preview

W [illegible] ext in Markdown

Tags

Create another task Duplicate to multiple projects

Save or cancel

Color: Yellow

Assignee: Unassigned Me

Column: Ready

Priority: 0

Due Date: 02/10/2025 19:37

Start Date: 02/10/2025 19:37

Original estimate hours

Time spent hours

Complexity

Reference

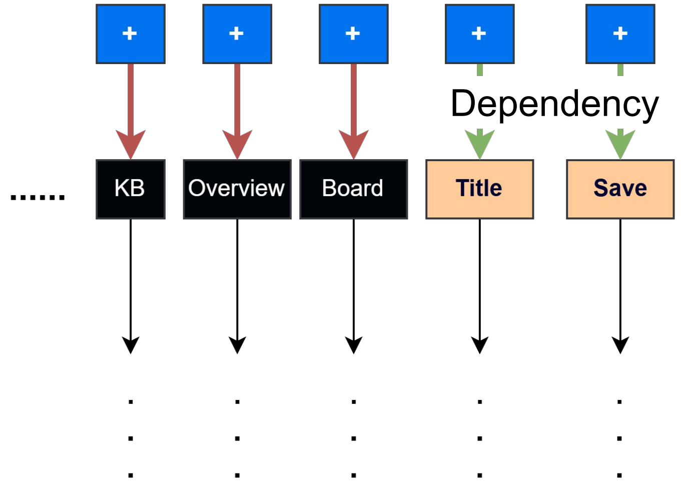
# Dependency Tracking

We consider dependencies to exist among elements when interaction with an element change the status of some web elements.

We design the scanner to keep track of these dependencies

- The “Title” and “Save” elements should follow the “+” elements instead of others

**Enable the scanner to quickly generate sequences with good order**



## Vulnerability Detections

Integrate IDOR and XSS vulnerability Detectors into EvoCrawl (Details are in the Paper)

Support future integration of other Vulnerability Detector

## Experiment Setup

Code Coverage

- Evaluate EvoCrawl's performance by comparing it with three state-of-the-art scanners: **BlackWidow, JAK, and CrawlJAX.**

Form Submissions (Measure how many states are explored):

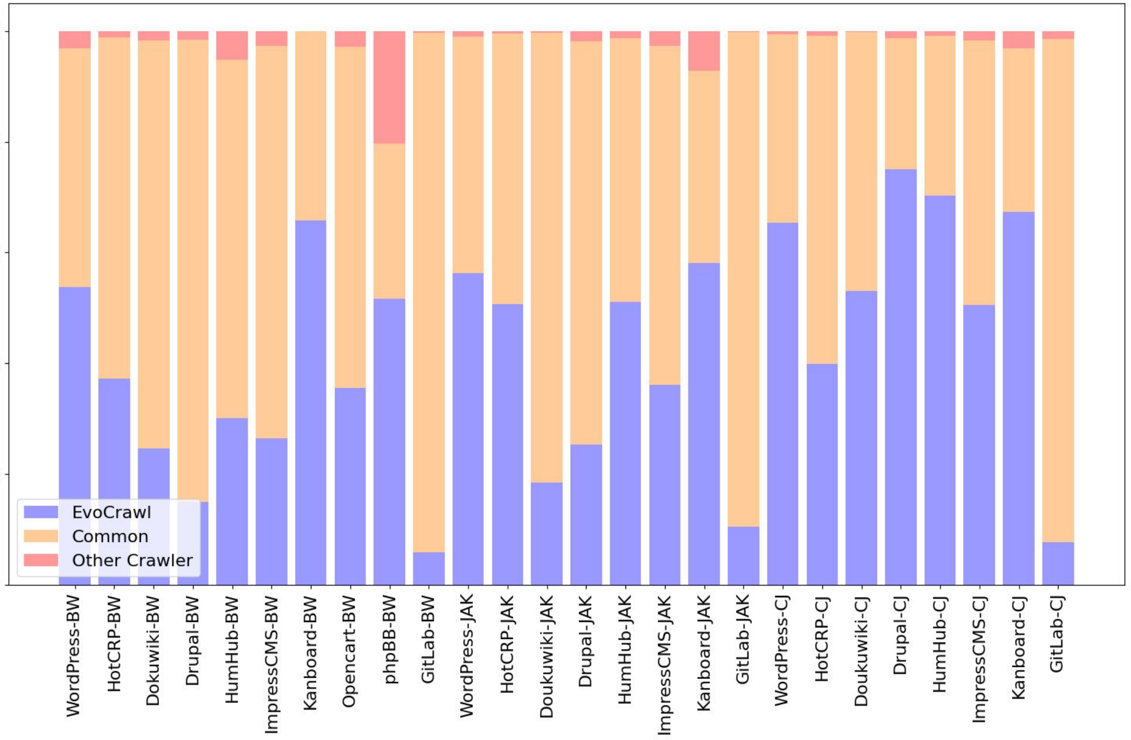
- Calculate the total count of HTML forms that have been submitted.
- **Compared with BlackWidow**

Vulnerability Detection

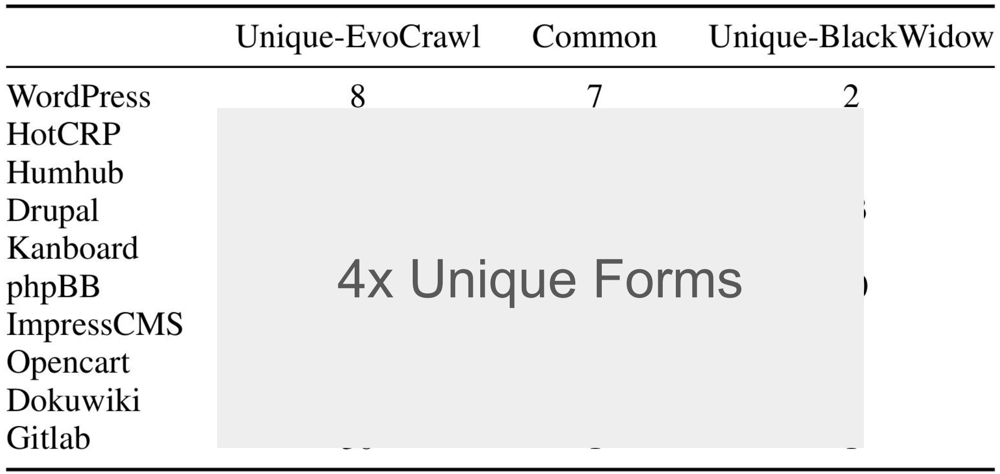
- Collect the number of vulnerabilities

## Code Coverage

Cover 1.5x lines of code



## Form Submissions



| | Unique-EvoCrawl | Common | Unique-BlackWidow |
|---|---|---|---|
| WordPress | 8 | 7 | 2 |
| HotCRP | [illegible] | [illegible] | [illegible] |
| Humhub | [illegible] | [illegible] | [illegible] |
| Drupal | [illegible] | [illegible] | [illegible] |
| Kanboard | [illegible] | [illegible] | [illegible] |
| phpBB | [illegible] | [illegible] | [illegible] |
| ImpressCMS | [illegible] | [illegible] | [illegible] |
| Opencart | [illegible] | [illegible] | [illegible] |
| Dokuwiki | [illegible] | [illegible] | [illegible] |
| Gitlab | [illegible] | [illegible] | [illegible] |

4x Unique Forms

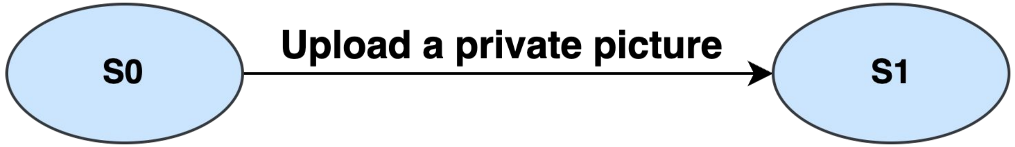
The count of submitted forms: Unique forms submitted by EvoCrawl, Common forms submitted by both crawlers and Unique forms submitted by BlackWidow (Details are in the Paper)

## IDOR Vulnerabilities Found

- Gitlab: One ajax endpoint that reveals all user’s information including avatar URL, username and states
- ImpressCMS:
  - One endpoint that allows attacker to force browsing to private images
  - One endpoint that allows attacker to force browsing to other user’s personal page

## IDOR Vulnerabilities Found

- Gitlab: One ajax endpoint that reveals all user’s information including avatar URL, username and states
- ImpressCMS:
    - **One endpoint that allows attacker to force browsing to private images**
    - One endpoint that allows attacker to force browsing to other user’s personal page



## XSS vulnerability Detector

**Humhub:** One injection point that allows the website owner to inject a custom script for tracking page statistics (Not a bug).

**Wordpress:** Two Stored XSS. (Acknowledged but not fixed because falls outside their security policy)

**HotCRP:**

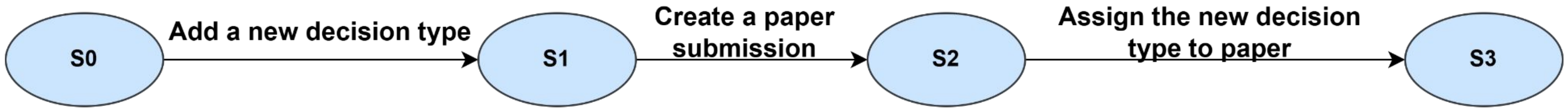
- One stored XSS vulnerability which has been acknowledged and fixed (fixed).
- One reflected XSS vulnerability which cannot be exploited by attackers as it is only visible to admin users and protected by a CSRF token. (Not acknowledged)

**Kanboard:** One stored XSS vulnerability that has been acknowledged and fixed.

# XSS vulnerability Detector

**HotCRP:**

- One stored XSS vulnerability which has been acknowledged and fixed (fixed).
- For each step, the scanner need to find the right sequence of interactions


S0
Add a new decision type
S1
Create a paper submission
S2
Assign the new decision type to paper
S3

## Conclusion

- Using an evolutionary search algorithm with dependency tracking enables a scanner to transition the application into diverse states, thus achieves greater code coverage
- Evolutionary search can guide the crawler toward favorable objectives, such as form submission, inputs injection, and elements that triggered JS events.

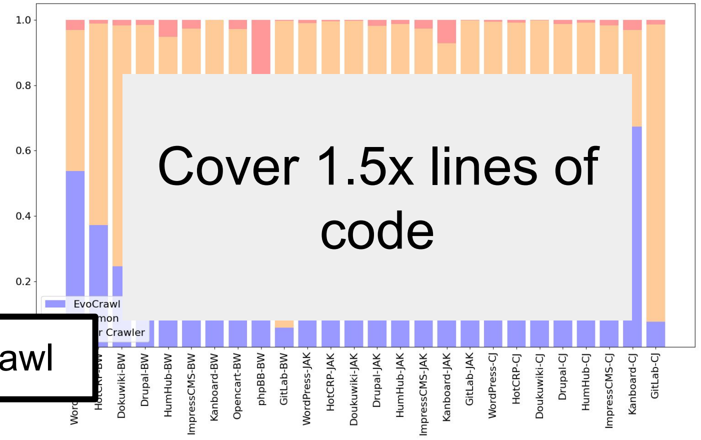
# Conclusion

Evolutionary Search

Dependency Tracking

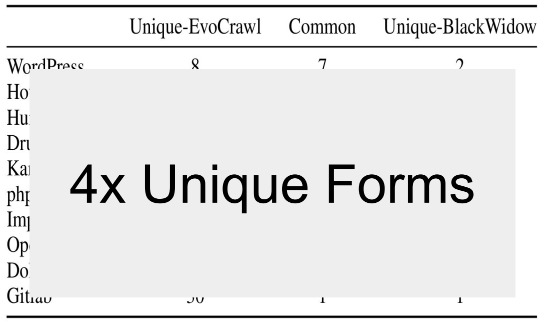
Main Idea

https://github.com/dlgroupuoft/evocrawl



Cover 1.5x lines of code

Found: 3 IDOR and 5 XSS



| | Unique-EvoCrawl | Common | Unique-BlackWidow |
|---|---|---|---|
| WordPress | 8 | 7 | 2 |
| Ho | | | |
| Hu | | | |
| Dr | | | |
| Ka | | | |
| php | | | |
| Im | | | |
| Op | | | |
| Do | | | |
| Gitlab | 30 | 1 | 1 |

4x Unique Forms

# Thank you!