

# Untangling the Knot: Recommending Component Refactorings

## Problem

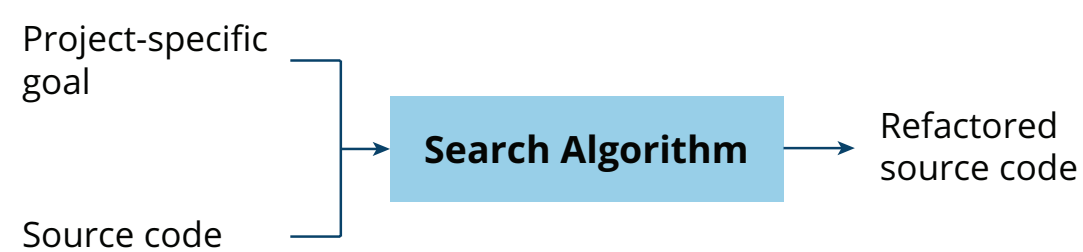
To quickly deliver new capabilities and take advantage of new technologies, DoD needs the ability to efficiently restructure software for common scenarios like

- migrating a capability to the cloud
- harvesting a component for reuse
- replacing a proprietary component

One recent anecdote estimates the effort to isolate a mission capability from the underlying hardware platform at 14,000 staff hours just for development.

## Solution

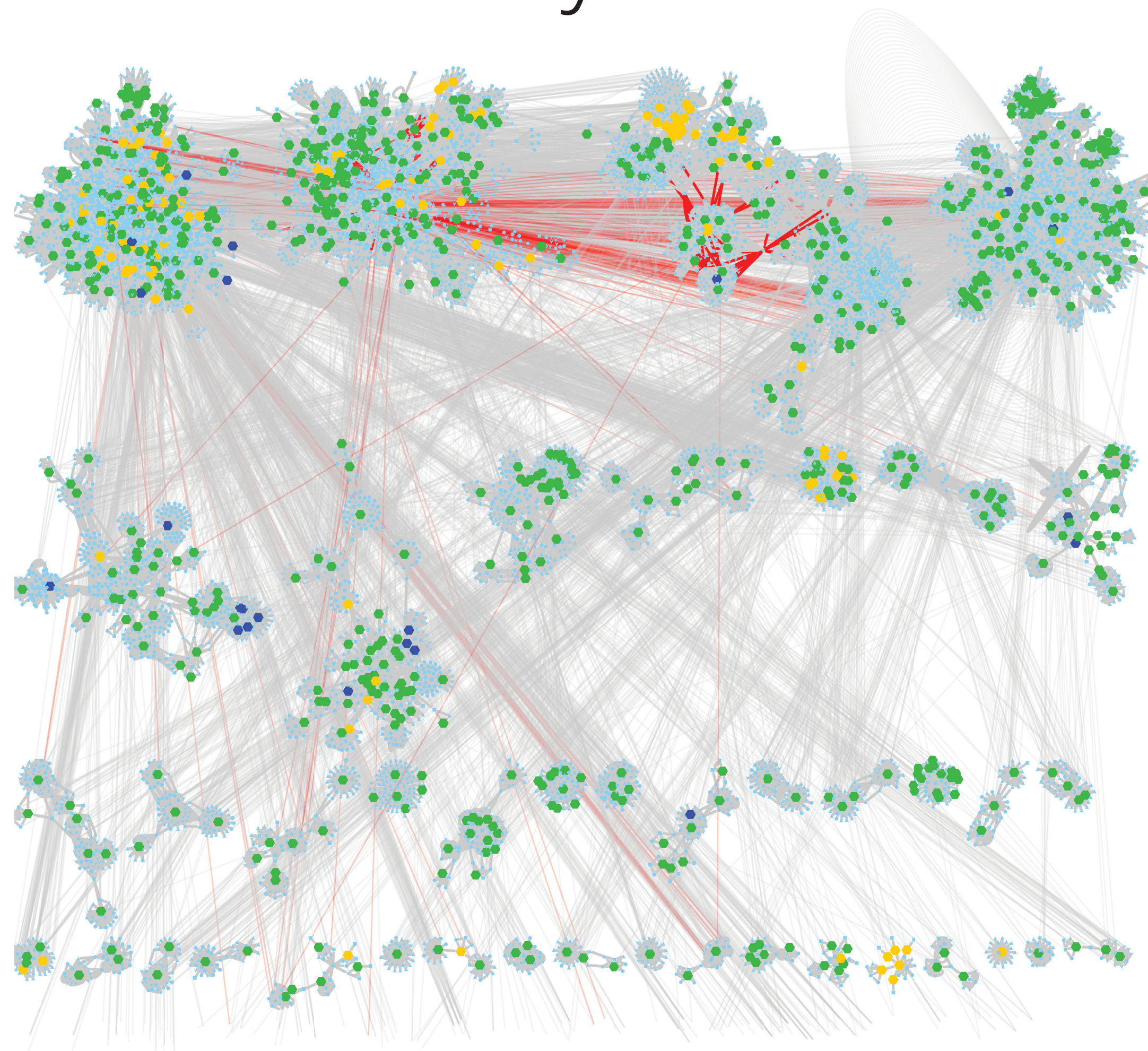
- Create a graph representation of software structure.
- Formalize refactorings as operations over the graph.
- Define fitness functions that evaluate the quality of candidate solutions.
- Apply search-based algorithms to resolve as many problematic couplings as possible while optimizing for multiple objectives.



## Intended Impact (FY19–21)

- Our refactoring recommendations outperform those based only on quality metrics, reducing problematic couplings by at least 75%.
- Developers accept recommendations.
- Our automation reduces the time to restructure software to 1/3 of the time compared to manual effort.

Automation using **search-based refactoring** can reduce the time required to restructure software to **1/3 of the time** it takes to do so manually.



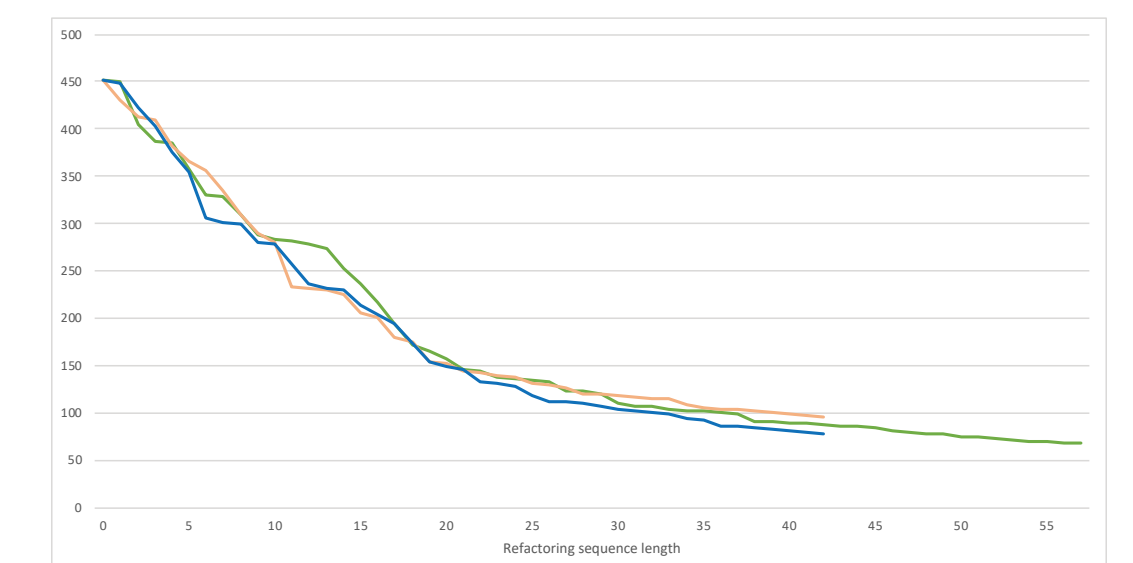
Our prototype automatically identifies problematic couplings that increase complexity and hinder software restructuring.

Project	Scenario	Problematic Couplings - Relation Type						Total
		CALLS	IMPLS	INHERITS	READS	USES_TYPE	WRITES	
MissionPlanner	Logger_A	515	0	1	982	255	403	2156
MissionPlanner	Logger_D	25	0	0	9	5	1	40
MissionPlanner	Radio_A	135	0	0	103	30	43	310
MissionPlanner	UI_A	2557	2	2	7269	2085	1493	13408
Duplicati	Logging_D	448	4	2	114	28	0	596
Duplicati	Server_A	105	3	0	235	56	52	451
Duplicati	Server_D	65	4	0	320	22	24	435
ConvNetSharp	GPU_D	529	0	1	495	384	7	1416
SharpCaster	Activity_D	10	0	0	13	3	0	26
eShopOnContaine	Eventbus_D	28	0	0	29	19	0	76
eShopOnContaine	Ordering_A	57	18	31	142	78	51	377
mRemoteNG	Putty_D	6	0	6	32	8	12	64
mRemoteNG	Rdp_A	45	1	1	218	4	16	285
mRemoteNG	Rdp_D	5	0	0	42	30	1	78
		4530	32	44	10003	3007	2103	

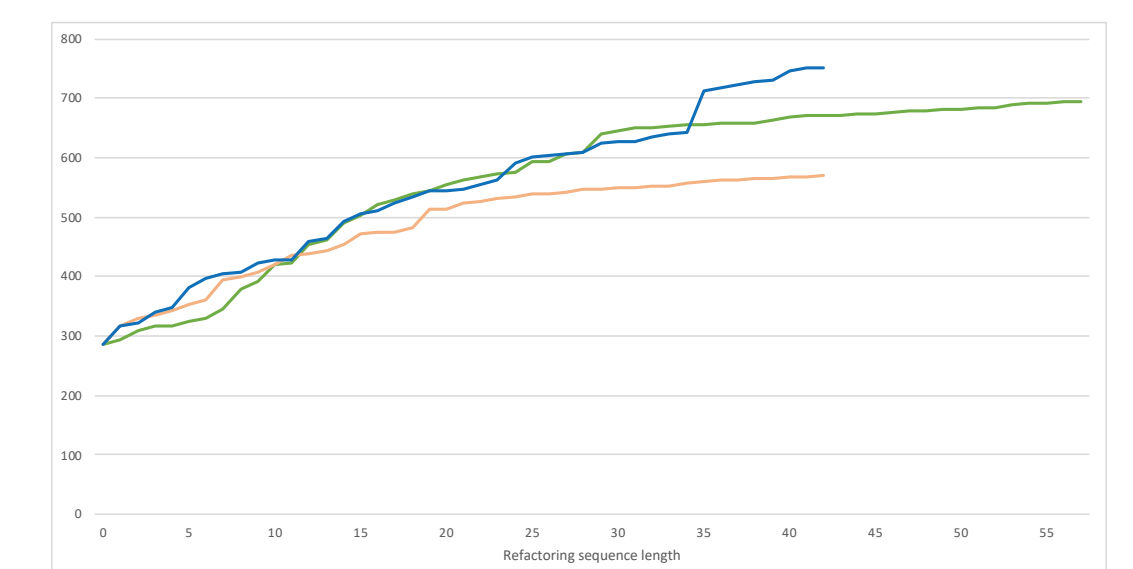
Source: All project data from [github.com/open-source](https://github.com/open-source)

Automated search finds sequences of refactorings that collectively solve as much of the project-specific goal as possible.

## Number of Problematic Couplings



## Amount of Code Included in Harvest



DoD programs with C# implementations can use our interim results now to

- size proposed changes and help prioritize software for migration
- provide input to cost analyses