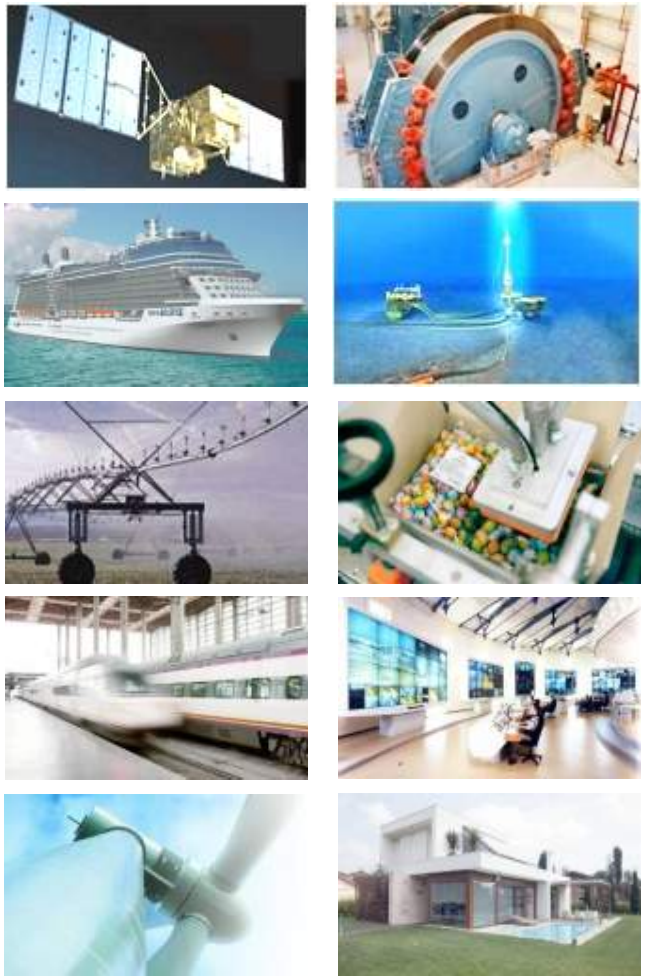
Code Change Impact Analysis for Testing Configurable Software Systems

Mithun Acharya
ABB Corporate Research
Raleigh NC USA

ABB: A power and automation company



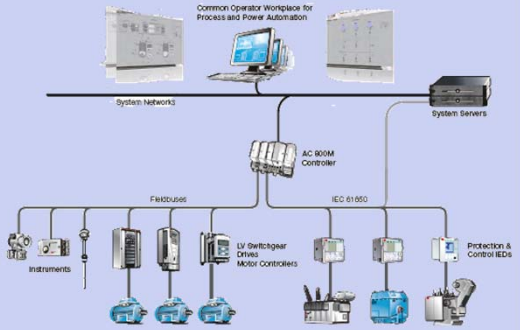
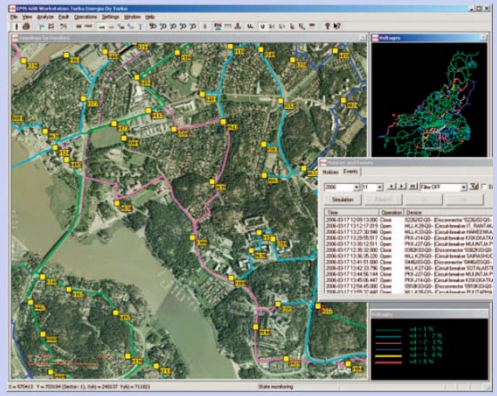
>125 years, >100 nations, ~150,000 employees

Power products and electronics, Control Systems, Robotics, Smart Grid, Renewable Energy, …

ABB Corporate Research
Industrial Software Systems (ISS) research group

USA
Germany
Switzerland
Poland
Sweden
India
China

Software in ABB

Hardware with software inside	Software with few hardware components	Pure Software
		

Software Evolution: A CSS constantly changes

pgmvpt_prev.c

```
1  PRIVATE void updateppc(
2    PGM_TASK *task,         /* Task descriptor */
3    VPT *vpt)             /* Viewport */
4  {
5      PPCLST *ppc;
6      BOOL first;
7
8      PGM_CHKNOP(vpt->window.typ != PGM_WINDOW_NONE);
9
10     if (vpt->window.typ == PGM_WINDOW_CUTBUF)
11         return;
12
13     if (vpt->tppc)
14     {
15         while (ppc != vpt->tppc->next)
16         {
17             if (checkchild(task, ppc, &first))
18                 ppc = ppc->next;
19             else if (!first)
20             {
21
22                 PGM_CHKNOP(vpt->tppc);
23                 break;
24             }
25             else
26             {
27                 vpt->tppc = nextline(task, ppc, &vpt->tppcbot);
28                 if (!vpt->tppc)
29                     vpt->tppc = setppcwin(task, vpt->tppc,
                                      FALSE, &vpt->tppcbot);
30                 break;
31             }
32         }
33         (void) limitppc(task, vpt);
34     }
35     else
36         vpt->tppc = setppcwin(task, vpt->window, TRUE,
                                                  &vpt->tppcbot);
```

pgmvpt_current.c

```
1  PRIVATE void updateppc(
2    PGM_TASK *task,         /* Task descriptor */
3    VPT *vpt)             /* Viewport */
4  {
5      PPCLST *ppc;
6      BOOL first;
7
8      PGM_CHKNOP(vpt->window.typ != PGM_WINDOW_NONE);
9
10     if (vpt->window.typ == PGM_WINDOW_CUTBUF)
11         return;
12
13     if (vpt->tppc)
14     {
15         ppc = vpt->tppcbot;
16         while (ppc != vpt->tppc->next)
17         {
18             if (checkchild(task, ppc, &first))
19                 ppc = ppc->next;
20             else if (first)
21             {
22                 vpt->tppc = firstline(task, ppc,
                                                  &vpt->tppcbot);
23                 PGM_CHKNOP(vpt->tppc);
24                 break;
25             }
26             else
27             {
28                 vpt->tppc = nextline(task, ppc, &vpt->tppcbot);
29                 if (!vpt->tppc)
30                     vpt->tppc = setppcwin(task, vpt->window,
                                      FALSE, &vpt->tppcbot);
31                 break;
32             }
33         }
34         (void) limitppc(task, vpt);
35     }
36     else
37         vpt->tppc = setppcwin(task, vpt->window, TRUE,
                                                  &vpt->tppcbot);
```

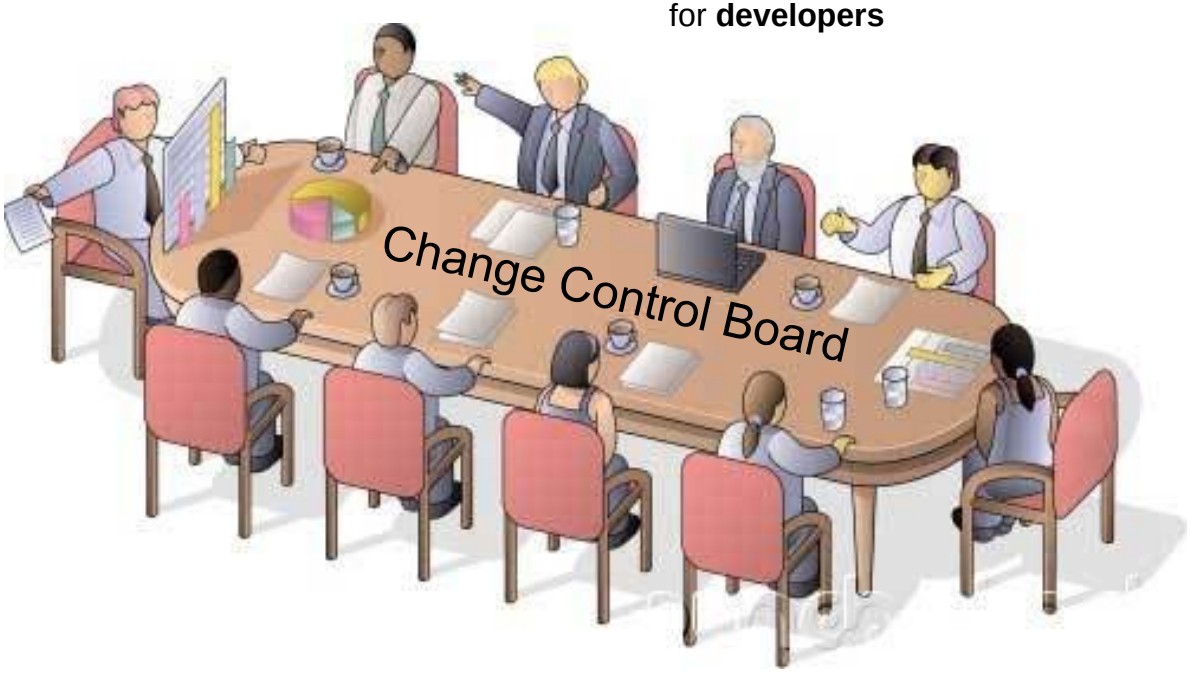
CHANGE

Hundreds of such changes committed daily

Change Control Board meetings

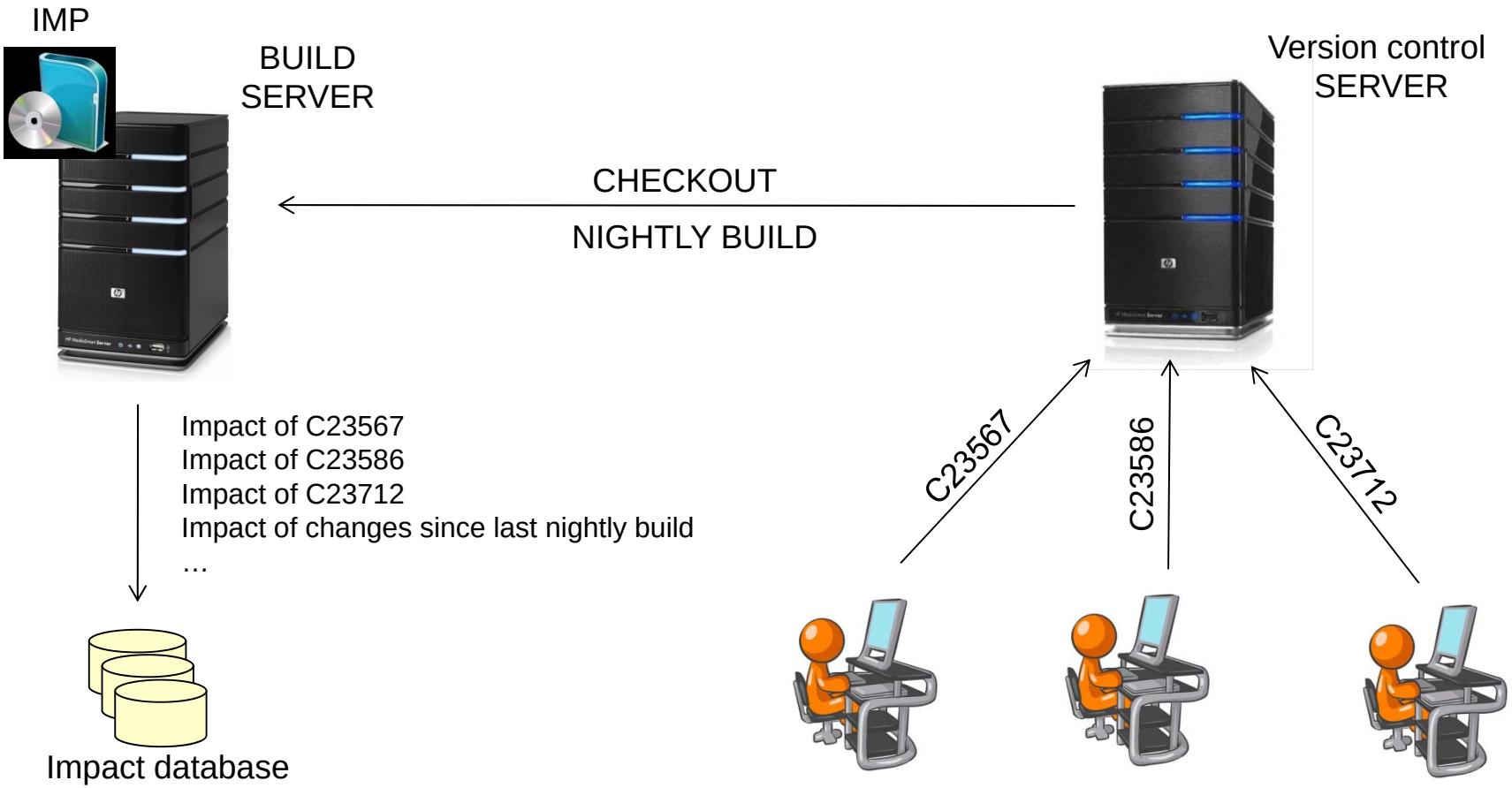
Change impact visualizations for **managers** for decision making

Change impact at the code level for **developers**



Unit/module-level change impact for **testers**

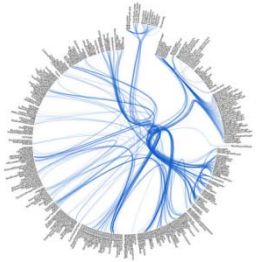
Imp: Code change impact analysis for C/C++ programs



Quantifiable risk/cost analysis of changes to CSS

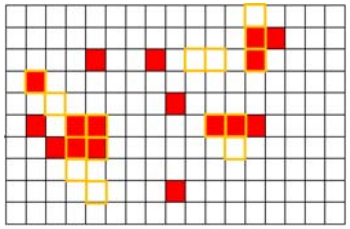
Automated Dependency Analysis

Will changes to foo.c, affect Bob's module? Dependency analysis

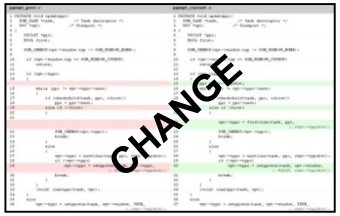


Automated Risk/Cost Analysis

3 days to release!!! Should I implement this feature or bug fix?

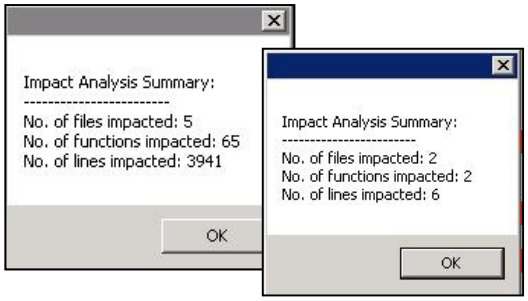


Overlay change impact with risky areas in code



Automated What-If Analysis

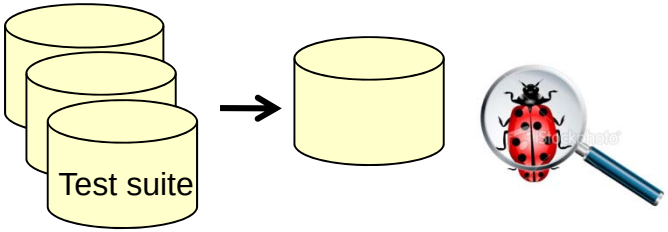
What is the 'best' way to fix this bug or implement that new feature?



3941 lines vs. 6 lines

Automated Regression Testing

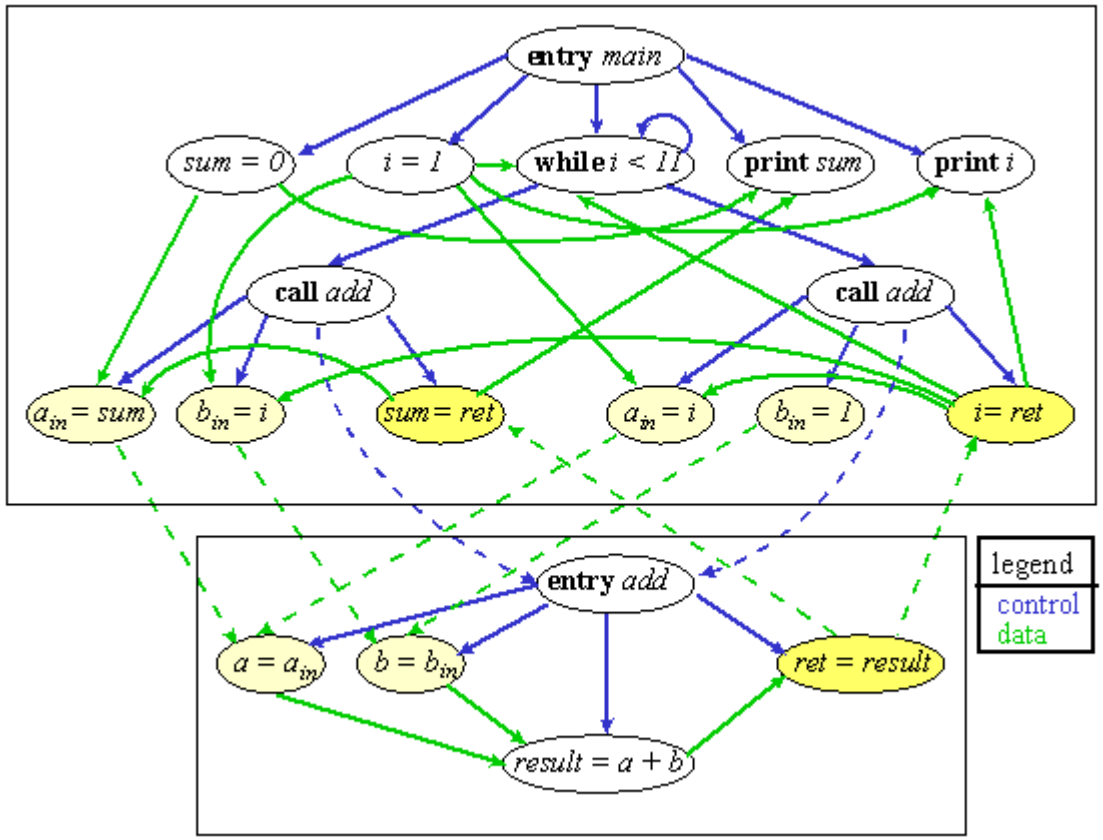
Should I re-run *ALL* of my test suite for this change? New tests required?



Program and System Dependence Graphs for Slicing

```
void main() {
    int i = 1;
    int sum = 0;
    while (i<11) {
        sum = add(sum, i);
        i = add(i, 1);
    }
    printf("sum = %d\n", sum);
    printf("i = %d\n", i);
}


static int add(int a, int b)
{
    return(a+b);
}
```

Program Dependence Graph (PDG) for `main`



PDG for `add`

System Dependence Graph (SDG)

Code/Image Source: GrammaTech

Making impact analysis practical and useful

Making impact analysis practical and useful

Mithun Acharya, Brian Robinson. *Practical Change Impact Analysis based on Static Program Slicing for Industrial Software Systems*. **ICSE 2011 SEiP, FSE 2012 (tool demo)**

Mithun Acharya, Xiao Qu, Brian Robinson. *Cross-System Change Impact Analysis Using Test Cases*. *Under submission*.

Scaling beyond million lines

Making impact analysis practical and useful

Mithun Acharya, Brian Robinson. *Practical Change Impact Analysis based on Static Program Slicing for Industrial Software Systems*. **ICSE 2011 SEiP, FSE 2012 (tool demo)**

Mithun Acharya, Xiao Qu, Brian Robinson. *Cross-System Change Impact Analysis Using Test Cases*. *Under submission*.

Scaling beyond million lines

Xiao Qu, Mithun Acharya, Brian Robinson. *Configuration Selection Using Code Change Impact Analysis for Regression Testing*. **ICSM 2012**

Xiao Qu, Mithun Acharya, Brian Robinson. *Impact Analysis of Configuration Changes for Test Case Selection*. **ISSRE 2011**

Testing configurable systems

Making impact analysis practical and useful

Mithun Acharya, Brian Robinson. *Practical Change Impact Analysis based on Static Program Slicing for Industrial Software Systems*. **ICSE 2011 SEiP, FSE 2012 (tool demo)**

Mithun Acharya, Xiao Qu, Brian Robinson. *Cross-System Change Impact Analysis Using Test Cases*. *Under submission*.

} Scaling beyond million lines

Xiao Qu, Mithun Acharya, Brian Robinson. *Configuration Selection Using Code Change Impact Analysis for Regression Testing*. **ICSM 2012**

Xiao Qu, Mithun Acharya, Brian Robinson. *Impact Analysis of Configuration Changes for Test Case Selection*. **ISSRE 2011**

} Testing configurable systems

Tingting Yu, Xiao Qu, Mithun Achayra, Gregg Rothermel. *Oracle-Based Regression Test Selection*. *Under submission*. → Regression test selection

What configurations should we select for retesting?

Mithun Acharya, Brian Robinson. *Practical Change Impact Analysis based on Static Program Slicing for Industrial Software Systems*. **ICSE 2011 SEiP, FSE 2012 (tool demo)**

Mithun Acharya, Xiao Qu, Brian Robinson. *Cross-System Change Impact Analysis Using Test Cases. Under submission*.

} Scaling beyond million lines

Xiao Qu, Mithun Acharya, Brian Robinson. *Configuration Selection Using Code Change Impact Analysis for Regression Testing*. **ICSM 2012**

Xiao Qu, Mithun Acharya, Brian Robinson. *Impact Analysis of Configuration Changes for Test Case Selection.* **ISSRE 2011**

} Testing configurable systems

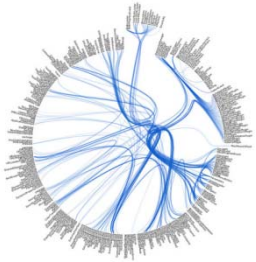
Tingting Yu, Xiao Qu, Mithun Achayra, Gregg Rothermel. *Oracle-Based Regression Test Selection. Under submission*.

→ Regression test selection

Regression testing of CSS with code change impact analysis

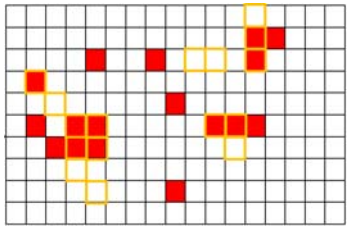
Automated Dependency Analysis

Will changes to foo.c, affect Bob's module? Dependency analysis

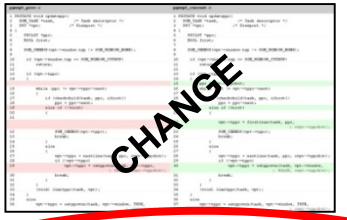


Automated Risk/Cost Analysis

3 days to release!!! Should I implement this feature or bug fix?

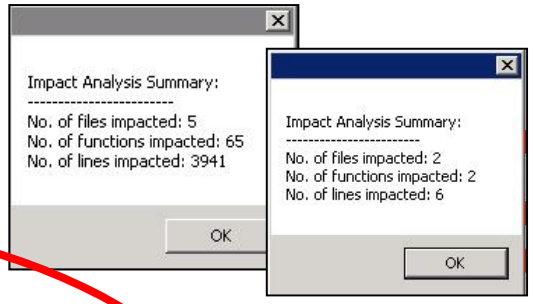


Overlay change impact with risky areas in code



Automated What-If Analysis

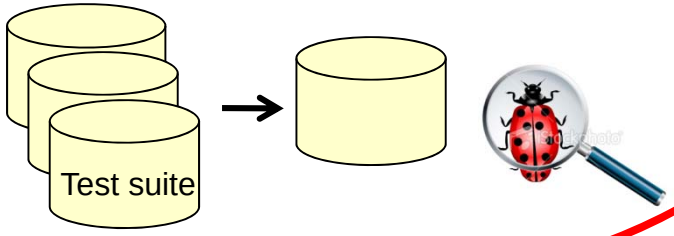
What is the 'best' way to fix this bug or implement that new feature?



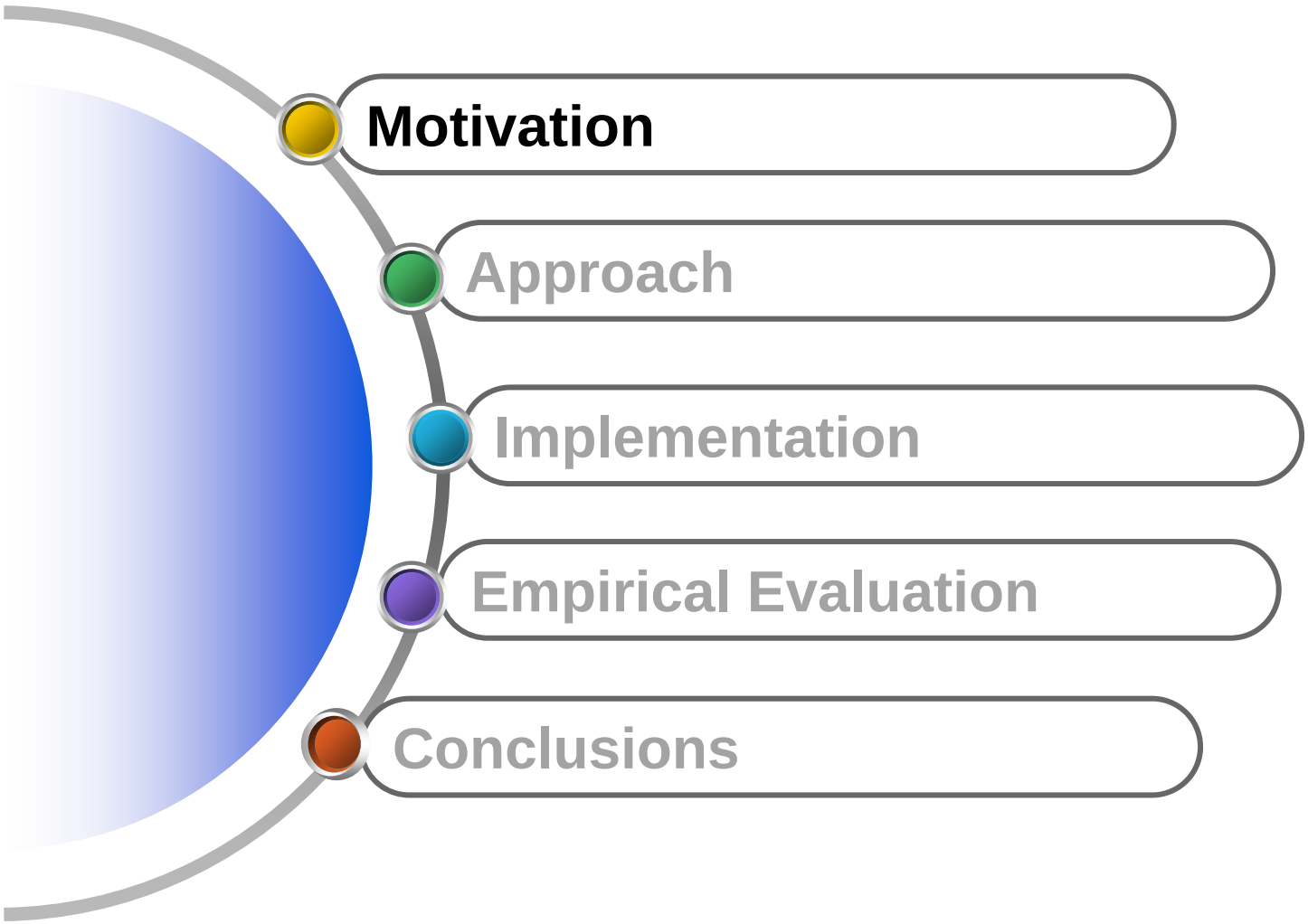
3941 lines vs. 6 lines

Automated Regression Testing

Should I re-run *ALL* of my test suite for this change? New tests required?

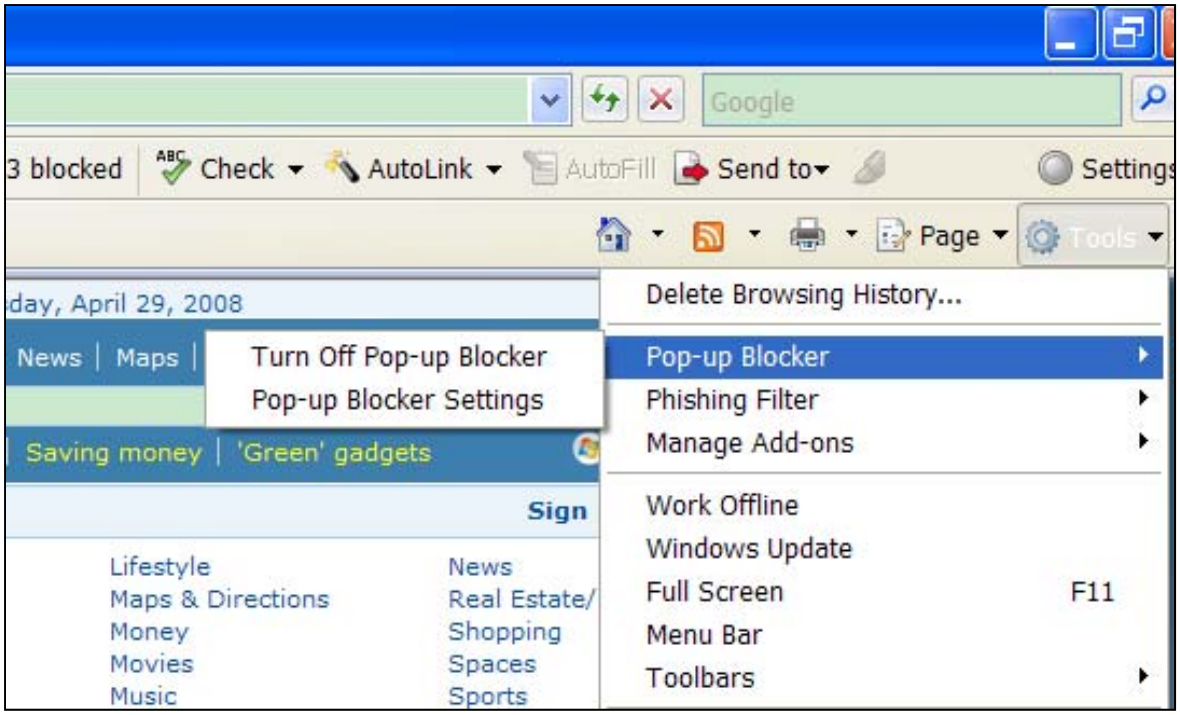


Outline


Motivation
Approach
Implementation
Empirical Evaluation
Conclusions

Configurable Software Systems

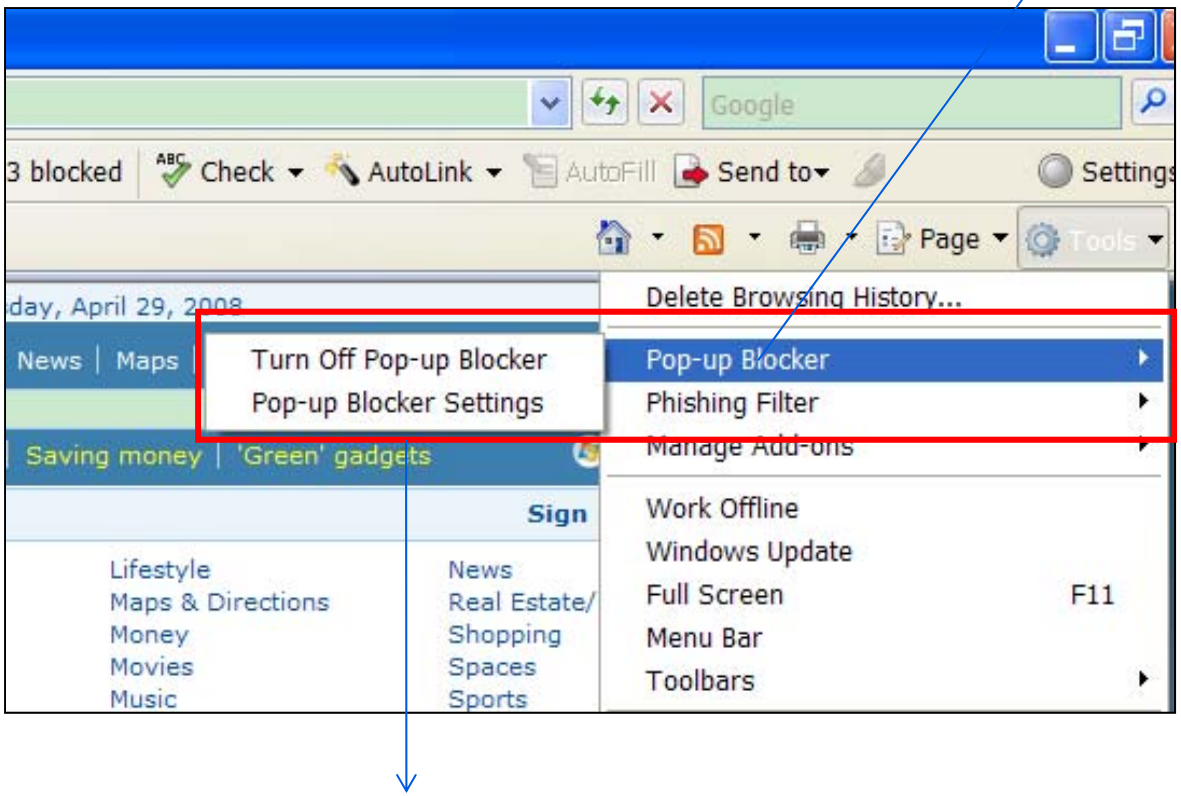
- Software that can be customized through a set of options
- Example: Internet Explorer



Configurable Software Systems

- Software that can be customized through a set of options
- Example: Internet Explorer

Configurable option: "**Pop-up Blocker**"

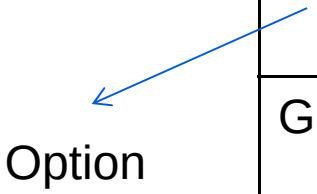


Values: **ON, OFF**

Internet Explorer Configurations

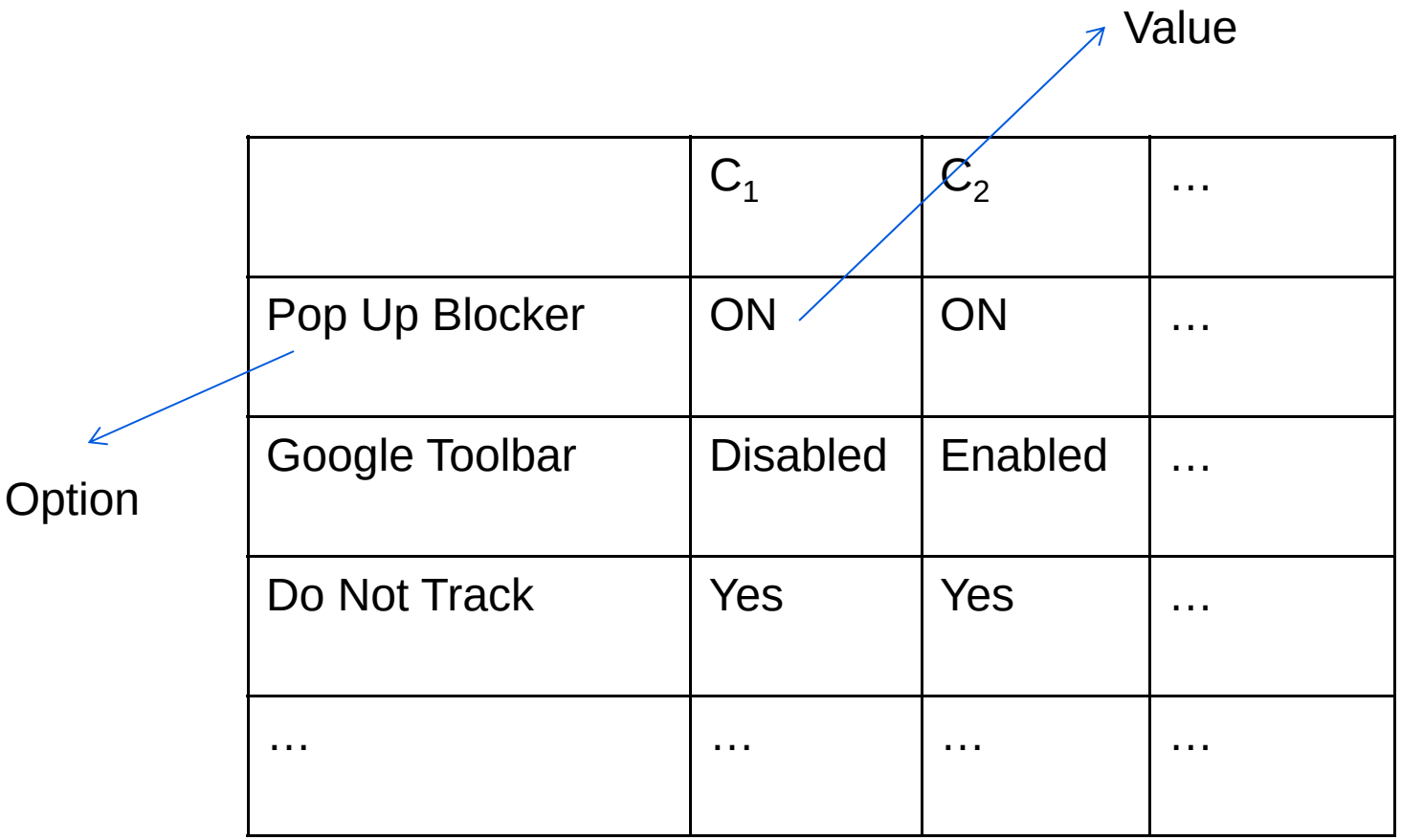
	C_1	C_2	…
Pop Up Blocker	ON	ON	…
Google Toolbar	Disabled	Enabled	…
Do Not Track	Yes	Yes	…
…	…	…	…

Internet Explorer Configurations



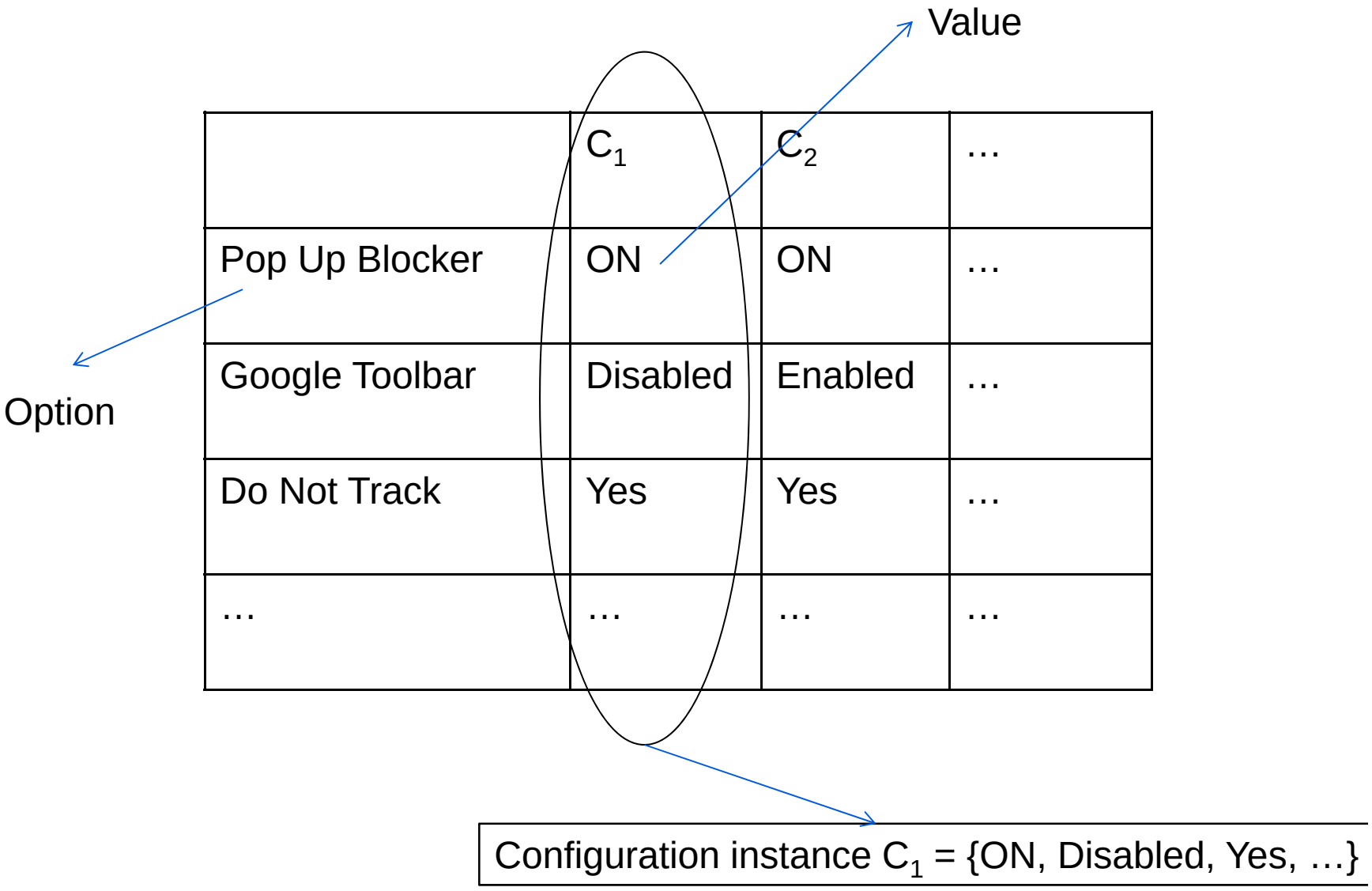
	C_1	C_2	…
Pop Up Blocker	ON	ON	…
Google Toolbar	Disabled	Enabled	…
Do Not Track	Yes	Yes	…
…	…	…	…

Internet Explorer Configurations

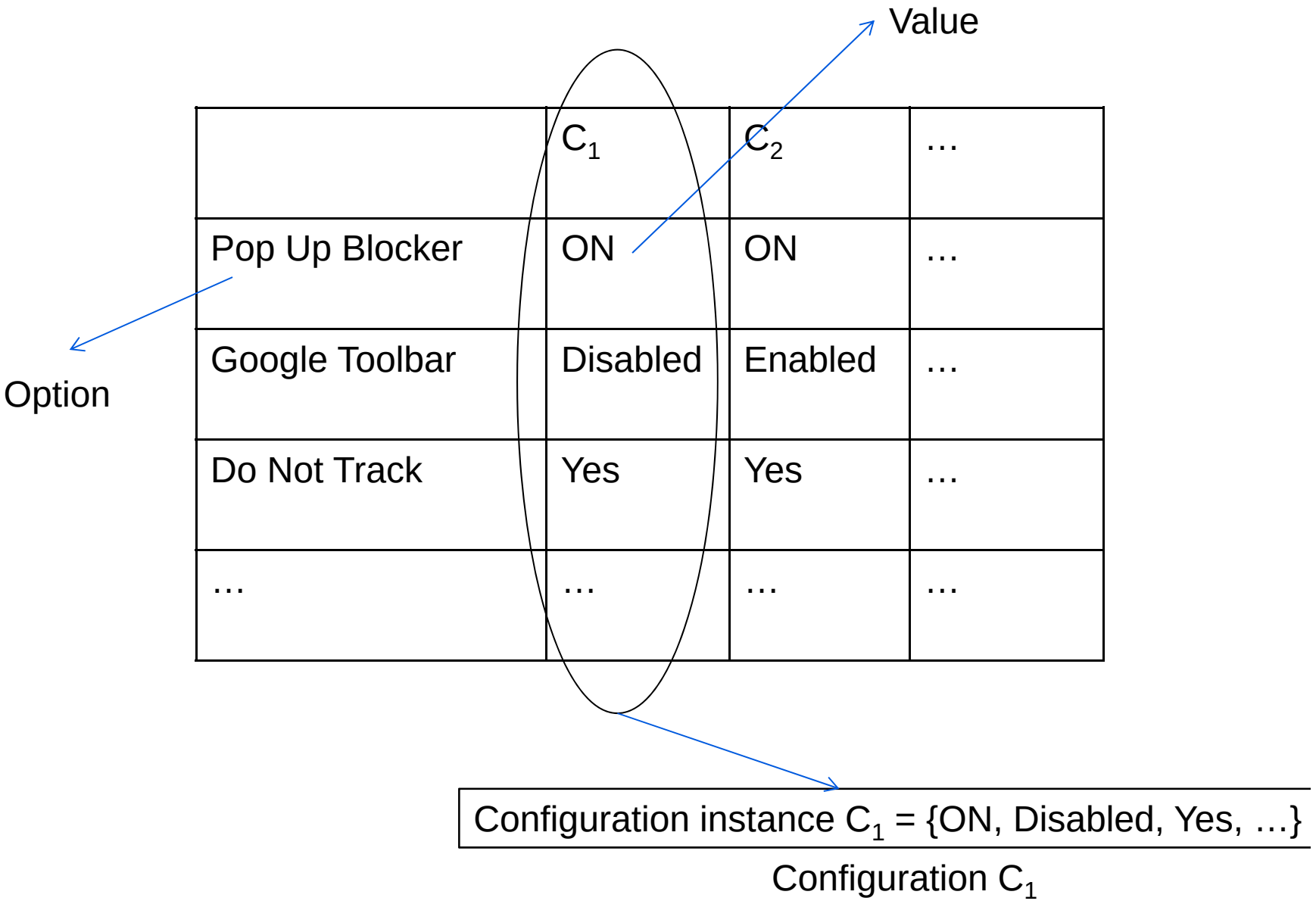


	C_1	C_2	…
Pop Up Blocker	ON	ON	…
Google Toolbar	Disabled	Enabled	…
Do Not Track	Yes	Yes	…
…	…	…	…

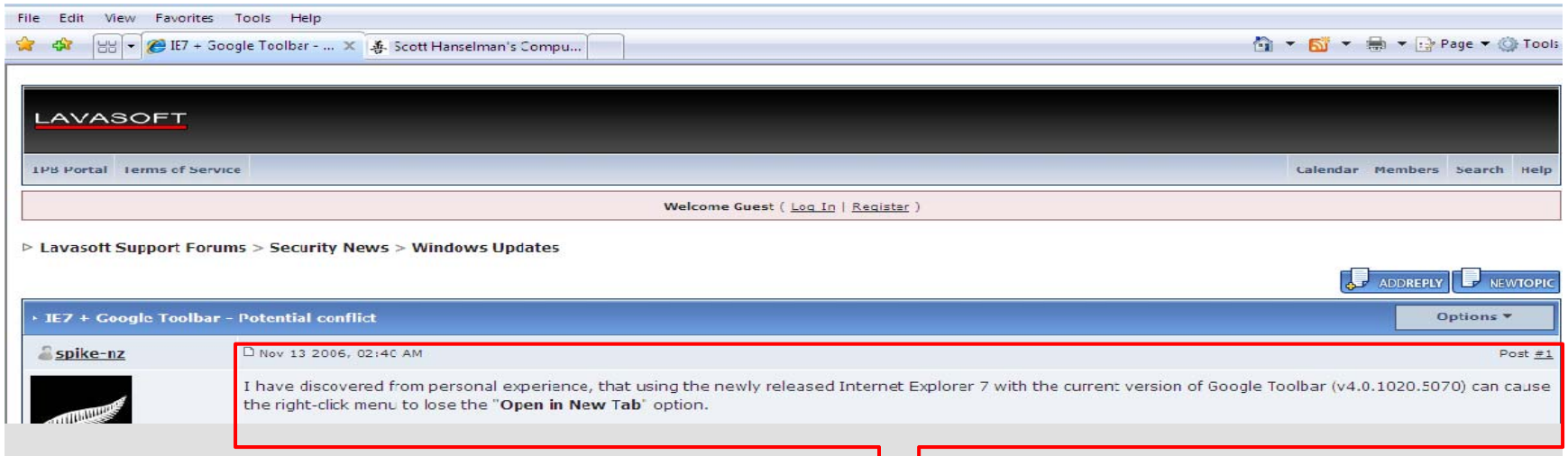
Internet Explorer Configurations



Internet Explorer Configurations

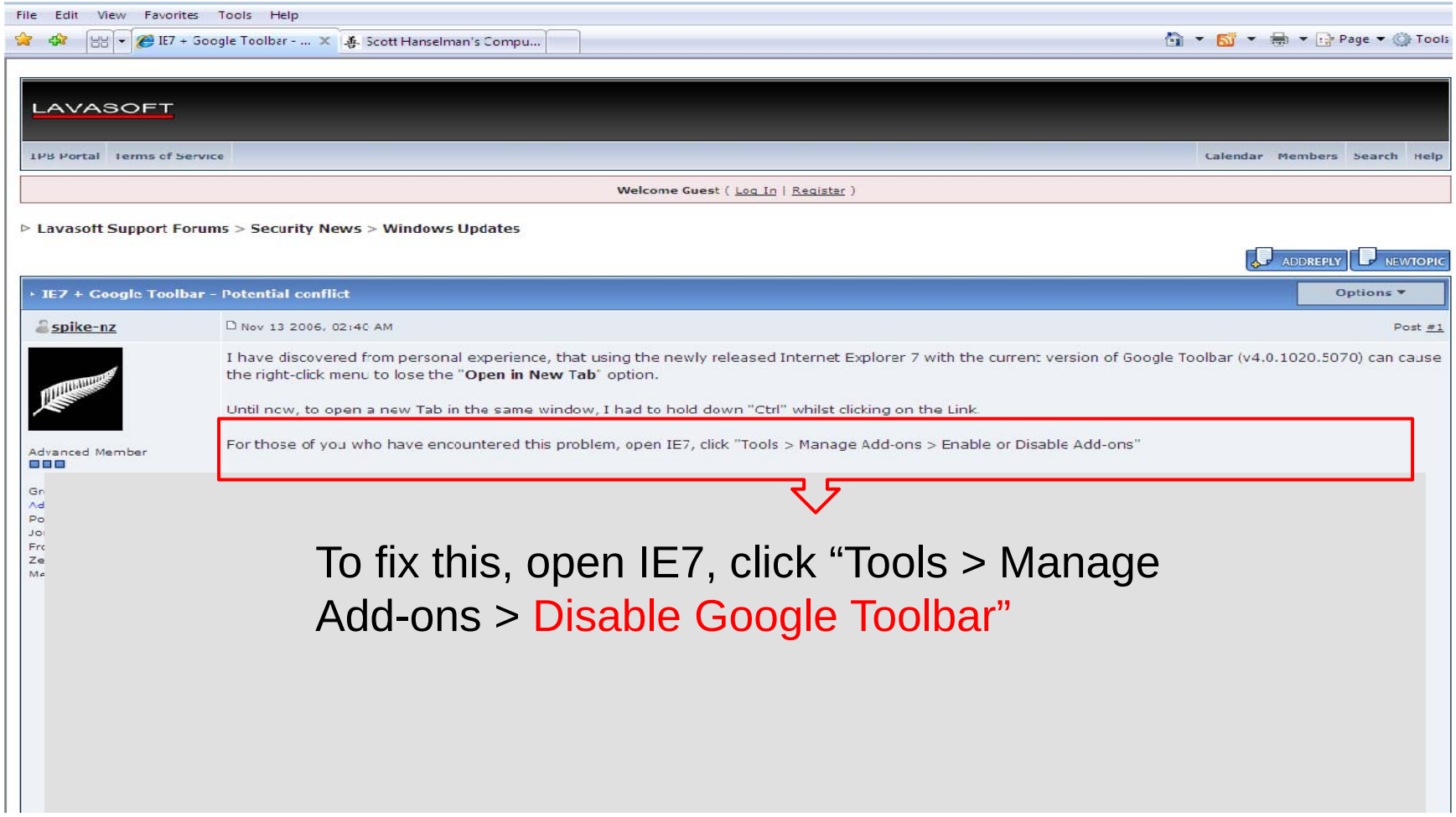


Impact of Configurations on System Behavior
Faulty System Behavior under Certain Configurations



I have discovered that using the newly released IE 7 with **Google Toolbar** = **Enabled** can cause the right-click menu to lose the “Open In New Tab” option

Impact of Configuration on System Behavior



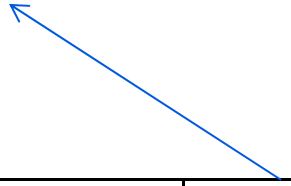
To fix this, open IE7, click “Tools > Manage Add-ons > Disable Google Toolbar”

Test Case: Open IE7, Right Click a link on webpage

	C_1	C_2	
Pop Up Blocker	ON	ON	…
Google Toolbar	Disabled	Enabled	…
Do Not Track	Yes	Yes	…
…	…	…	…

Test Case: Open IE7, Right Click a link on webpage

Test: PASS



	C_1	C_2	
Pop Up Blocker	ON	ON	…
Google Toolbar	Disabled	Enabled	…
Do Not Track	Yes	Yes	…
…	…	…	…

Test Case: Open IE7, Right Click a link on webpage

Test: PASS

Test: FAIL
No “Open in New Tab”

	C_1	C_2	
Pop Up Blocker	ON	ON	…
Google Toolbar	Disabled	Enabled	…
Do Not Track	Yes	Yes	…
…	…	…	…

Test Case: Open IE7, Right Click a link on webpage

Test: PASS

Test: FAIL
No “Open in New Tab”

	C_1	C_2	
Pop Up Blocker	ON	ON	…
Google Toolbar	Disabled	Enabled	…
Do Not Track	Yes	Yes	…
…	…	…	…

A test case that passes with one configuration may fail with another

Configurations control system execution

IE7

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token

  no_login_required

  cattr_writer :cache_timeout
  def self.cache_timeout
    @@cache_timeout ||= 5.minutes
  end

  def show_page
    url = params[:url]
    if Array === url
      url = url.join('/')
    else
      url = url.to_s
    end

    if @page = find_page(url)
      process_page(@page)
      set_cache_control
      @performed_render ||= true
    else
      render :template => 'site/not_found', :status => 404
    end
  rescue Page::MissingRootPageError
    redirect_to welcome_url
  end

  private
    def set_cache_control
      if (request.head? || request.get?) && @page.cache? && live?
        expires_in self.class.cache_timeout, :public => true, :private => false
      else
```

IE7

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token

  no_login_required

  cattr_writer :cache_timeout
  def self.cache_timeout
    @@cache_timeout ||= 5.minutes
  end

  def show_page
    url = params[:url]
    if Array === url
      url = url.join('/')
    else
      url = url.to_s
    end

    if @page = find_page(url)
      process_page(@page)
      set_cache_control
      @performed_render ||= true
    else
      render :template => 'site/not_found', :status => 404
    end
  rescue Page::MissingRootPageError
    redirect_to welcome_url
  end

  private
    def set_cache_control
      if (request.head? || request.get?) && @page.cache? && live?
        expires_in self.class.cache_timeout, :public => true, :private => false
      else
```

Configurations control system execution

IE7

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token

  no_login_required

  cattr_writer :cache_timeout
  def self.cache_timeout
    @@cache_timeout ||= 5.minutes
  end

  def show_page
    url = params[:url]
    if Array === url
      url = url.join('/')
    else
      url = url.to_s
    end

    if @page = find_page(url)
      process_page(@page)
      set_cache_control
      @performed_render ||= true
    else
      render :template => 'site/not_found', :status => 404
    end
  rescue Page::MissingRootPageError
    redirect_to welcome_url
  end

  private
    def set_cache_control
      if (request.head? || request.get?) && @page.cache? && live?
        expires_in self.class.cache_timeout, :public => true, :private => false
      else
```

IE7

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token

  no_login_required

  cattr_writer :cache_timeout
  def self.cache_timeout
    @@cache_timeout ||= 5.minutes
  end

  def show_page
    url = params[:url]
    if Array === url
      url = url.join('/')
    else
      url = url.to_s
    end

    if @page = find_page(url)
      process_page(@page)
      set_cache_control
      @performed_render ||= true
    else
      render :template => 'site/not_found', :status => 404
    end
  rescue Page::MissingRootPageError
    redirect_to welcome_url
  end

  private
    def set_cache_control
      if (request.head? || request.get?) && @page.cache? && live?
        expires_in self.class.cache_timeout, :public => true, :private => false
      else
```

C_1 = {ON, disabled, Yes, …}

Configurations control system execution

IE7

Test: PASS

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token

  no_login_required

  cattr_writer :cache_timeout
  def self.cache_timeout
    @@cache_timeout ||= 5.minutes
  end

  def show_page
    url = params[:url]
    if Array === url
      url = url.join('/')
    else
      url = url.to_s
    end

    if @page = find_page(url)
      process_page(@page)
      set_cache_control
      @performed_render ||= true
    else
      render :template => 'site/not_found', :status => 404
    end
  rescue Page::MissingRootPageError
    redirect_to welcome_url
  end

  private
    def set_cache_control
      if (request.head? || request.get?) && @page.cache? && live?
        expires_in self.class.cache_timeout, :public => true, :private => false
      else
```

C_1 = {ON, disabled, Yes, …}

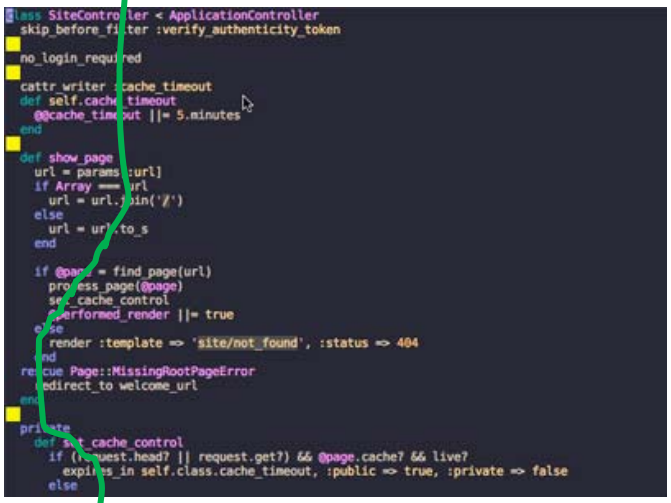
IE7

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token

  no_login_required

  cattr_writer :cache_timeout
  def self.cache_timeout
    @@cache_timeout ||= 5.minutes
  end

  def show_page
    url = params[:url]
    if Array === url
      url = url.join('/')
    else
      url = url.to_s
    end

    if @page = find_page(url)
      process_page(@page)
      set_cache_control
      @performed_render ||= true
    else
      render :template => 'site/not_found', :status => 404
    end
  rescue Page::MissingRootPageError
    redirect_to welcome_url
  end

  private
    def set_cache_control
      if (request.head? || request.get?) && @page.cache? && live?
        expires_in self.class.cache_timeout, :public => true, :private => false
      else
```

Configurations control system execution

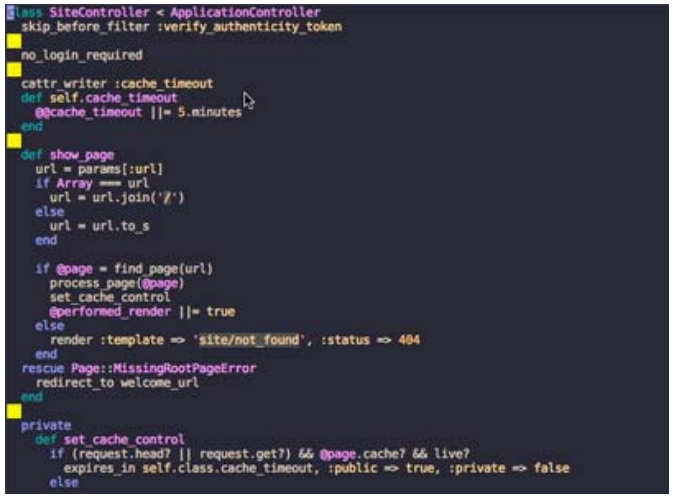
IE7

Test: PASS



```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token

  no_login_required

  cattr_writer :cache_timeout
  def self.cache_timeout
    @@cache_timeout ||= 5.minutes
  end

  def show_page
    url = params[:url]
    if Array === url
      url = url.join('/')
    else
      url = url.to_s
    end

    if @page = find_page(url)
      process_page(@page)
      set_cache_control
      @performed_render ||= true
    else
      render :template => 'site/not_found', :status => 404
    end
  rescue Page::MissingRootPageError
    redirect_to welcome_url
  end

  private
    def set_cache_control
      if (request.head? || request.get?) && @page.cache? && live?
        expires_in self.class.cache_timeout, :public => true, :private => false
      else
```

IE7



```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token

  no_login_required

  cattr_writer :cache_timeout
  def self.cache_timeout
    @@cache_timeout ||= 5.minutes
  end

  def show_page
    url = params[:url]
    if Array === url
      url = url.join('/')
    else
      url = url.to_s
    end

    if @page = find_page(url)
      process_page(@page)
      set_cache_control
      @performed_render ||= true
    else
      render :template => 'site/not_found', :status => 404
    end
  rescue Page::MissingRootPageError
    redirect_to welcome_url
  end

  private
    def set_cache_control
      if (request.head? || request.get?) && @page.cache? && live?
        expires_in self.class.cache_timeout, :public => true, :private => false
      else
```

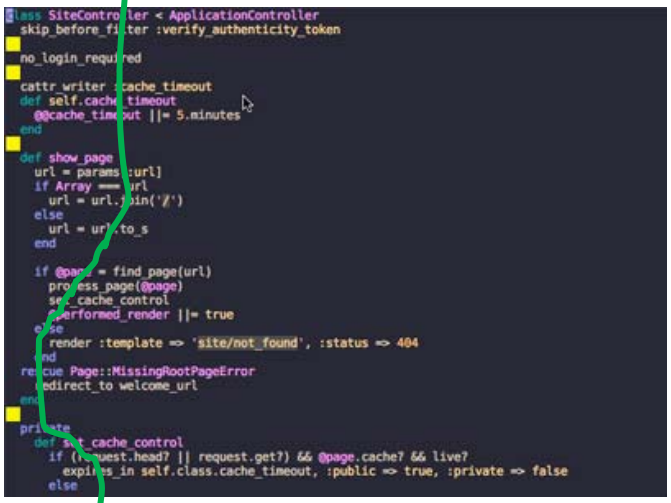
C_1 = {ON, disabled, Yes, …}

C_2 = {ON, enabled, Yes, …}

Configurations control system execution

IE7

Test: PASS



C_1 = {ON, disabled, Yes, …}

IE7

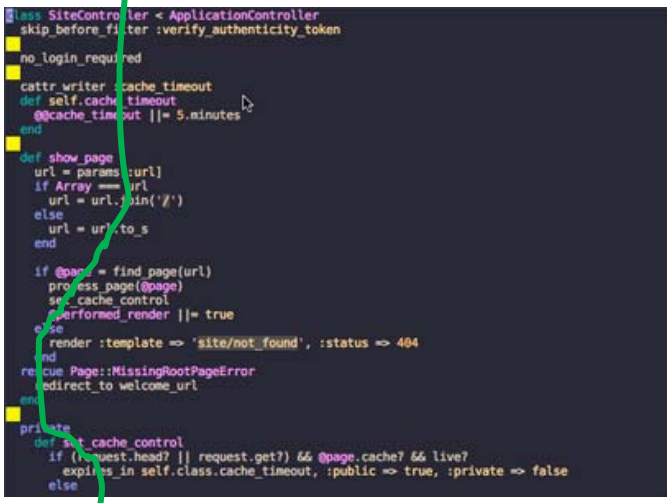
Test: FAIL

C_2 = {ON, enabled, Yes, …}

Configurations control system execution

IE7

Test: PASS



C_1 = {ON, disabled, Yes, …}

IE7

Test: FAIL

C_2 = {ON, enabled, Yes, …}

Can we statically approximate how configurations (options) control system execution?

Challenges for testing configurable systems

Challenges for testing configurable systems

n options  2^n configurations

Challenges for testing configurable systems

n options

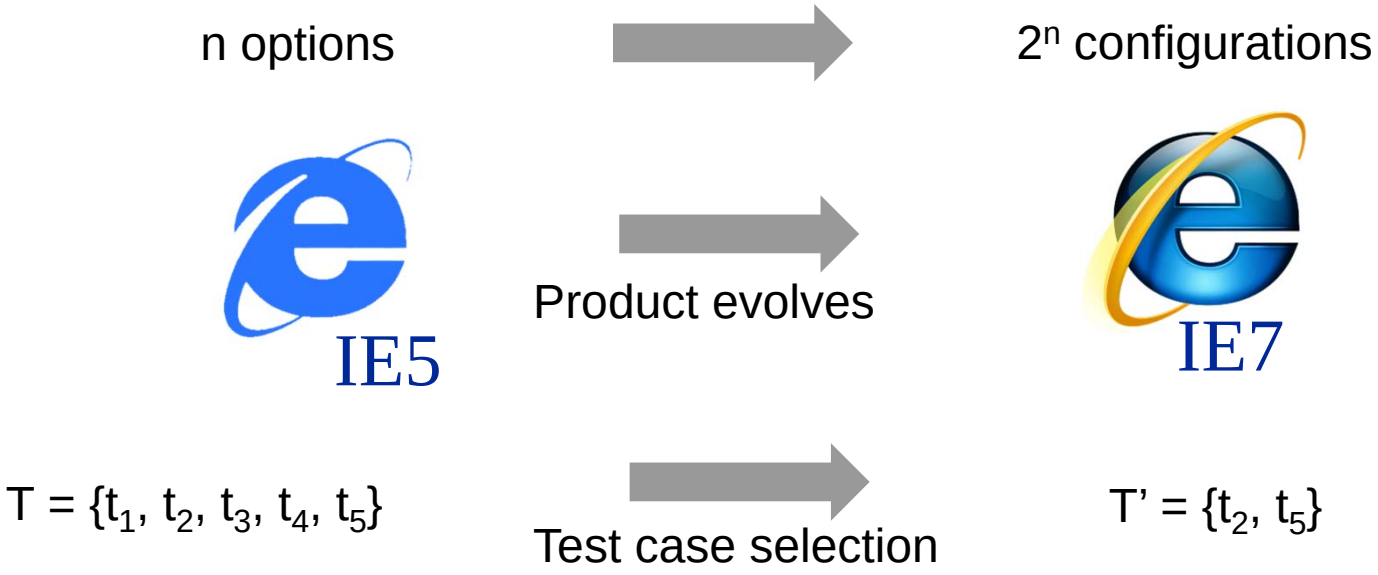
2^n configurations

IE5

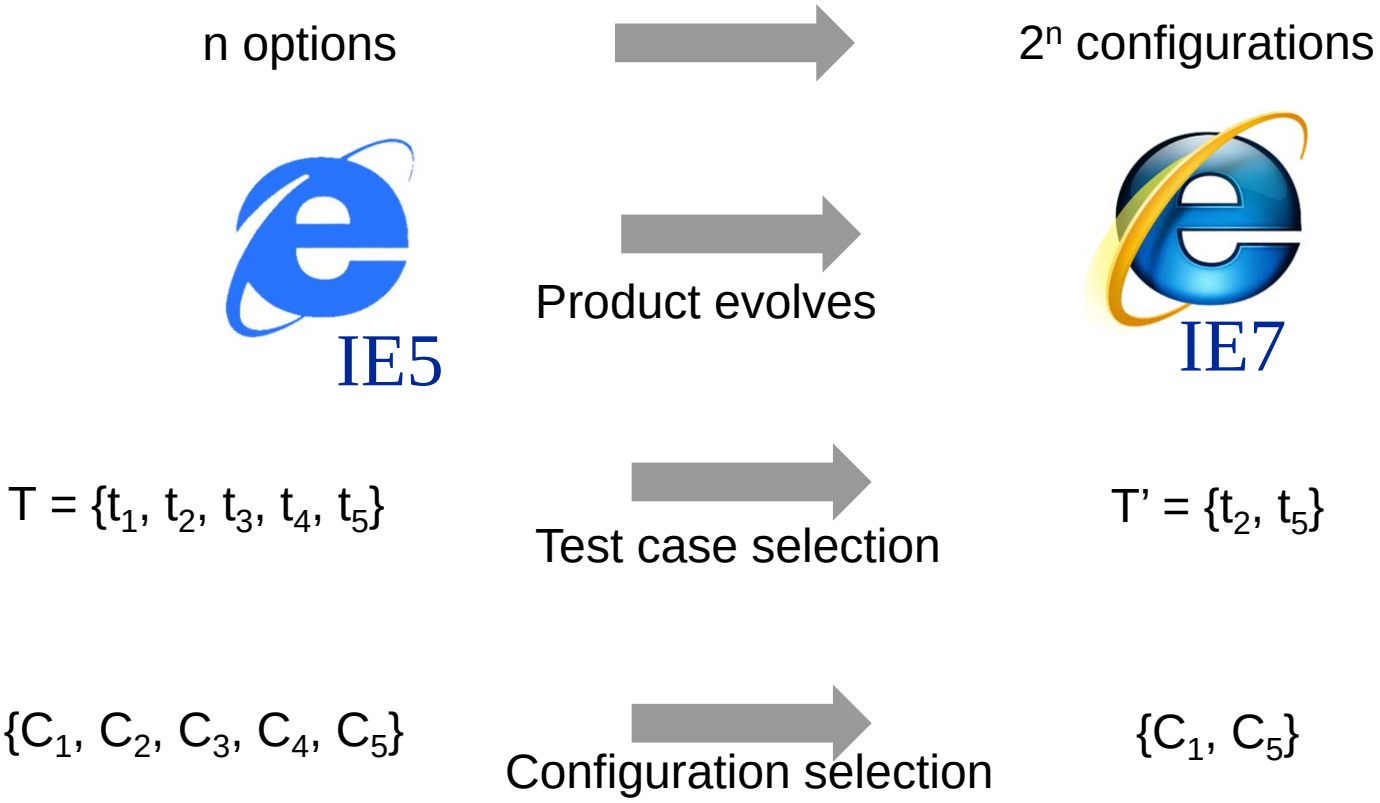
Product evolves

IE7

Challenges for testing configurable systems



Challenges for testing configurable systems



Configuration Sampling

Reducing the exponential number of configurations to a manageable size

Configuration Sampling

Reducing the exponential number of configurations to a manageable size

$\{C_1, C_2, C_3, C_4, C_5, \ldots\}$
Exponentially large set

$T = \{t_1, t_2, t_3, t_4, t_5\}$

Configuration Sampling

Reducing the exponential number of configurations to a manageable size

$\{C_1, C_2, C_3, C_4, C_5, \ldots\}$
Exponentially large set

Sampling

$\{C_1, C_3, C_4\}$
Manageable size set

$T = \{t_1, t_2, t_3, t_4, t_5\}$

Configuration Sampling

Reducing the exponential number of configurations to a manageable size

$\{C_1, C_2, C_3, C_4, C_5, \ldots\}$
Exponentially large set

Sampling

$\{C_1, C_3, C_4\}$
Manageable size set

$T = \{t_1, t_2, t_3, t_4, t_5\}$

No test case selection

$T' = \{t_1, t_2, t_3, t_4, t_5\}$

Configuration Sampling

Reducing the exponential number of configurations to a manageable size

$\{C_1, C_2, C_3, C_4, C_5, \ldots\}$
Exponentially large set

Sampling

$\{C_1, C_3, C_4\}$
Manageable size set

$T = \{t_1, t_2, t_3, t_4, t_5\}$

No test case selection

$T' = \{t_1, t_2, t_3, t_4, t_5\}$

Example: Configuration Interaction Testing (CIT)

Configuration Sampling

Reducing the exponential number of configurations to a manageable size

$\{C_1, C_2, C_3, C_4, C_5, \ldots\}$
Exponentially large set

Sampling

$\{C_1, C_3, C_4\}$
Manageable size set

$T = \{t_1, t_2, t_3, t_4, t_5\}$

No test case selection

$T' = \{t_1, t_2, t_3, t_4, t_5\}$

Example: Configuration Interaction Testing (CIT)

We choose to test IE7 only under sampled configurations C_1, C_3, and C_4
and for each configurations we test IE7 with all tests $\{t_1, t_2, t_3, t_4, t_5\}$

`vim`: A configurable system

`vim`: A configurable system

2^{90} configurations

`vim`: A configurable system

`vim`: A configurable system

Rerun the full test suite on each 60 configurations

`vim`: A configurable system

Rerun the full test suite on each 60 configurations

7 hours to execute the full test suite
Takes 7*60 = 420 hours (~2.5 weeks) to run all test cases under each configuration

`vim`: A configurable system

Rerun the full test suite on each 60 configurations

7 hours to execute the full test suite
Takes 7*60 = 420 hours (~2.5 weeks) to run all test cases under each configuration

Do we have to run all tests under each configuration?

Test case selection when configuration under test changes*

* Qu, Acharya, Robinson, “Impact analysis of configuration changes for test case selection”, ISSRE 2011

Test case selection when configuration under test changes*

* Qu, Acharya, Robinson, “Impact analysis of configuration changes for test case selection”, ISSRE 2011

Test case selection when configuration under test changes*

Source code DOES NOT change

Configuration under test changes

C_1 = {ON, Disabled, Yes, …} C_2 = {ON, Enabled, Yes, …}

* Qu, Acharya, Robinson, “Impact analysis of configuration changes for test case selection”, ISSRE 2011

Test case selection when configuration under test changes*

Source code DOES NOT change

Configuration under test changes

C_1 = {ON, Disabled, Yes, …} C_2 = {ON, Enabled, Yes, …}

What test cases should I re-run for the new configuration?

T = {t_1, t_2, t_3, t_4, t_5}

* Qu, Acharya, Robinson, “Impact analysis of configuration changes for test case selection”, ISSRE 2011

Test case selection when configuration under test changes*

Source code DOES NOT change

Configuration under test changes

C_1 = {ON, Disabled, Yes, …} C_2 = {ON, Enabled, Yes, …}

What test cases should I re-run for the new configuration?

T = {t_1, t_2, t_3, t_4, t_5} T' = {t_2, t_5}

* Qu, Acharya, Robinson, "Impact analysis of configuration changes for test case selection", ISSRE 2011

Test case selection when configuration under test changes*

Source code DOES NOT change

Configuration under test changes

C_1 = {ON, Disabled, Yes, …} C_2 = {ON, Enabled, Yes, …}

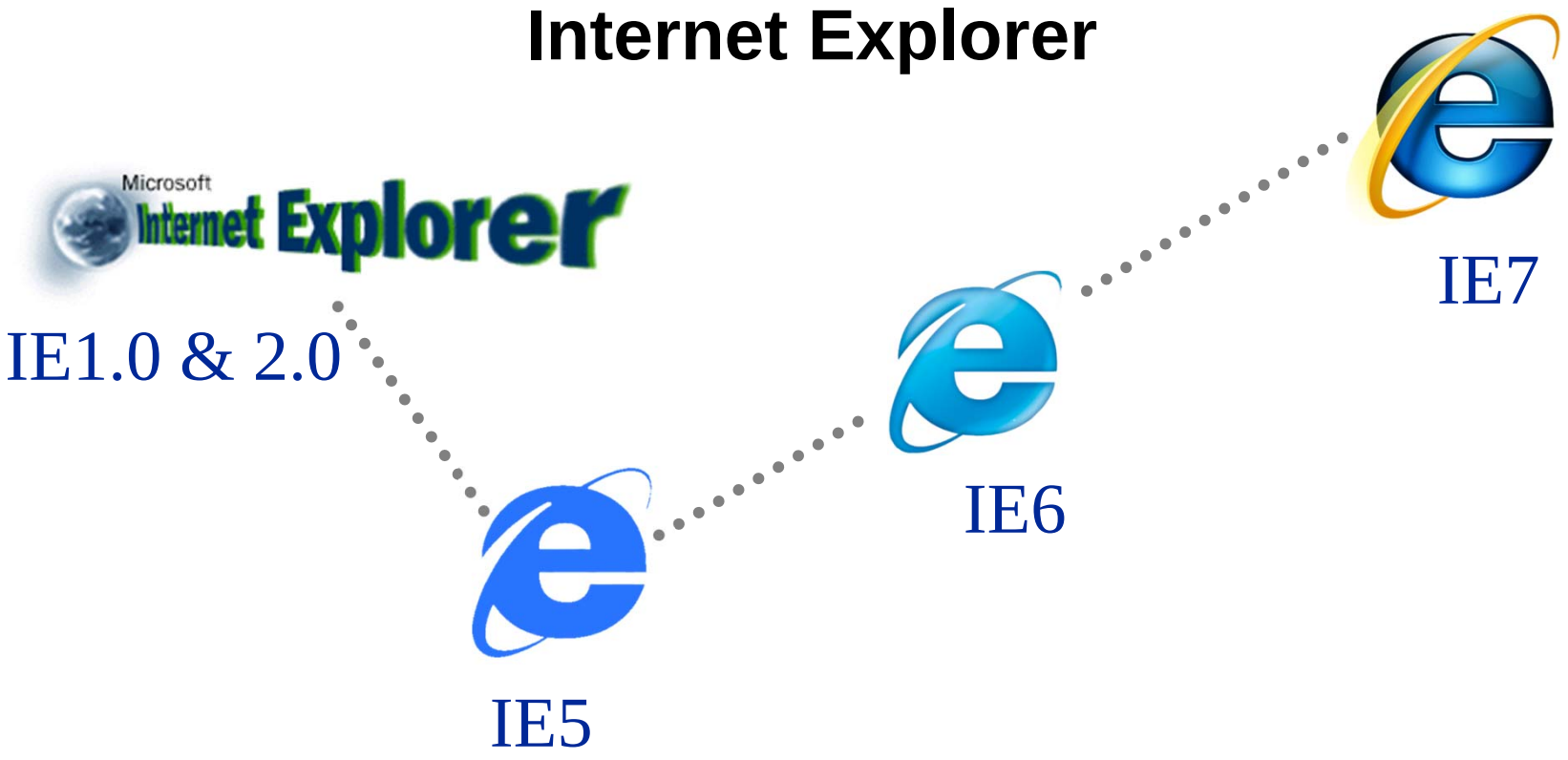
What test cases should I re-run for the new configuration?

T = {t_1, t_2, t_3, t_4, t_5} T' = {t_2, t_5}

For the ABB system analyzed, only about 20% of the tests had to be re-run for a configuration change

* Qu, Acharya, Robinson, "Impact analysis of configuration changes for test case selection", ISSRE 2011

Product Evolution



Configuration prioritization for regression testing*

*Qu, Cohen, Rothermel, “Configuration-aware regression testing: An empirical study of sampling and prioritization”, ISSTA 2008

Configuration prioritization for regression testing*

$\{C_1, C_2, C_3, C_4, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$

*Qu, Cohen, Rothermel, “Configuration-aware regression testing: An empirical study of sampling and prioritization”, ISSTA 2008

Configuration prioritization for regression testing*

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$

*Qu, Cohen, Rothermel, “Configuration-aware regression testing: An empirical study of sampling and prioritization”, ISSTA 2008

Configuration prioritization for regression testing*

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Prioritization → $\{C_1, C_4, C_{3.}\ ...\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$

*Qu, Cohen, Rothermel, “Configuration-aware regression testing: An empirical study of sampling and prioritization”, ISSTA 2008

Configuration prioritization for regression testing*

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Prioritization → $\{C_1, C_4, C_{3.}\ ...\}$

Reorder and run as many as you can

$T = \{t_1, t_2, t_3, t_4, t_5\}$

*Qu, Cohen, Rothermel, “Configuration-aware regression testing: An empirical study of sampling and prioritization”, ISSTA 2008

Configuration prioritization for regression testing*

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Prioritization → $\{C_1, C_4, C_{3.}\ ...\}$

Reorder and run as many as you can

$T = \{t_1, t_2, t_3, t_4, t_5\}$ No test case selection $T' = \{t_1, t_2, t_3, t_4, t_5\}$

*Qu, Cohen, Rothermel, “Configuration-aware regression testing: An empirical study of sampling and prioritization”, ISSTA 2008

Configuration prioritization for regression testing*

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Prioritization → $\{C_1, C_4, C_{3.\ ...}\}$

Reorder and run as many as you can

$T = \{t_1, t_2, t_3, t_4, t_5\}$ No test case selection $T' = \{t_1, t_2, t_3, t_4, t_5\}$

Increases rate of fault detection. But…

*Qu, Cohen, Rothermel, “Configuration-aware regression testing: An empirical study of sampling and prioritization”, ISSTA 2008

Configuration prioritization for regression testing*

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Prioritization → $\{C_1, C_4, C_{3.\ ...}\}$

Reorder and run as many as you can

$T = \{t_1, t_2, t_3, t_4, t_5\}$ No test case selection $T' = \{t_1, t_2, t_3, t_4, t_5\}$

Increases rate of fault detection. But…

Does not eliminate redundancy.
Does not detect all faults. Not safe.

*Qu, Cohen, Rothermel, “Configuration-aware regression testing: An empirical study of sampling and prioritization”, ISSTA 2008

Configuration prioritization for regression testing*

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Prioritization → $\{C_1, C_4, C_{3.\ ...}\}$

Reorder and run as many as you can

$T = \{t_1, t_2, t_3, t_4, t_5\}$ No test case selection $T' = \{t_1, t_2, t_3, t_4, t_5\}$

Increases rate of fault detection. But…

Does not eliminate redundancy.
Does not detect all faults. Not safe.

Can we select a subset of $\{C_1, C_2, C_3, C_4, C_5\}$ that is both non-redundant and safe?

*Qu, Cohen, Rothermel, "Configuration-aware regression testing: An empirical study of sampling and prioritization", ISSTA 2008

Configuration selection for regression testing (Focus of this talk)

Configuration selection for regression testing (Focus of this talk)

$\{C_1, C_2, C_3, C_4, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$

Configuration selection for regression testing (Focus of this talk)

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$

Configuration selection for regression testing (Focus of this talk)

Source code CHANGES

$\{C_1, C_2, C_3, C_4, C_5\}$ → Selection → $\{C_1, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$

Configuration selection for regression testing (Focus of this talk)

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Selection → $\{C_1, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$ No test case selection $T' = \{t_1, t_2, t_3, t_4, t_5\}$

Configuration selection for regression testing (Focus of this talk)

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Selection → $\{C_1, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$ No test case selection $T' = \{t_1, t_2, t_3, t_4, t_5\}$

$\{C_1, C_5\}$ is both safe (wrt retest-all configurations) and non redundant

State of the Art in Configurable System Testing

TABLE I. THE STATE OF THE ART IN CONFIGURABLE SYSTEM TESTING

Problems		Single Version Testing	Regression Testing
Configuration Level	Selection	[11][15][18][25]	**Focus of this talk**
	Prioritization	NA	[18]
Test Case Level	Selection	[17]	[22][23][a]
	Prioritization	NA	[21] [a]

State of the Art in Configurable System Testing

- Configuration sampling
- Single version
- No test case selection
- Example, CIT

TABLE I. THE STATE OF THE ART IN CONFIGURABLE SYSTEM TESTING

Problems		Single Version Testing	Regression Testing
Configuration Level	Selection	[11][15][18][25]	**Focus of this talk**
	Prioritization	NA	[18]
Test Case Level	Selection	[17]	[22][23][a]
	Prioritization	NA	[21] [a]

State of the Art in Configurable System Testing

- Configuration sampling
- Single version
- No test case selection
- Example, CIT

TABLE I. THE STATE OF THE ART IN CONFIGURABLE SYSTEM TESTING

Problems		Single Version Testing	Regression Testing
Configuration Level	Selection	[11][15][18][25]	**Focus of this talk**
	Prioritization	NA	[18]
Test Case Level	Selection	[17]	[22][23][a]
	Prioritization	NA	[21] [a]

- Test case selection [ISSRE '11]
- Single version
- Configuration under test changes
- Non-redundant
- Safe

State of the Art in Configurable System Testing

- Configuration sampling
- Single version
- No test case selection
- Example, CIT

TABLE I. THE STATE OF THE ART IN CONFIGURABLE SYSTEM TESTING

Problems		Single Version Testing	Regression Testing
Configuration Level	Selection	[11][15][18][25]	**Focus of this talk**
	Prioritization	NA	[18]
Test Case Level	Selection	[17]	[22][23][a]
	Prioritization	NA	[21] [a]

- Test case selection [ISSRE '11]
- Single version
- Configuration under test changes
- Non-redundant
- Safe

- Configuration prioritization [ISSTA '08]
- Source code changes
- Regression Testing
- No test case selection
- Redundant
- Not safe

State of the Art in Configurable System Testing

- Configuration sampling
- Single version
- No test case selection
- Example, CIT

TABLE I. THE STATE OF THE ART IN CONFIGURABLE SYSTEM TESTING

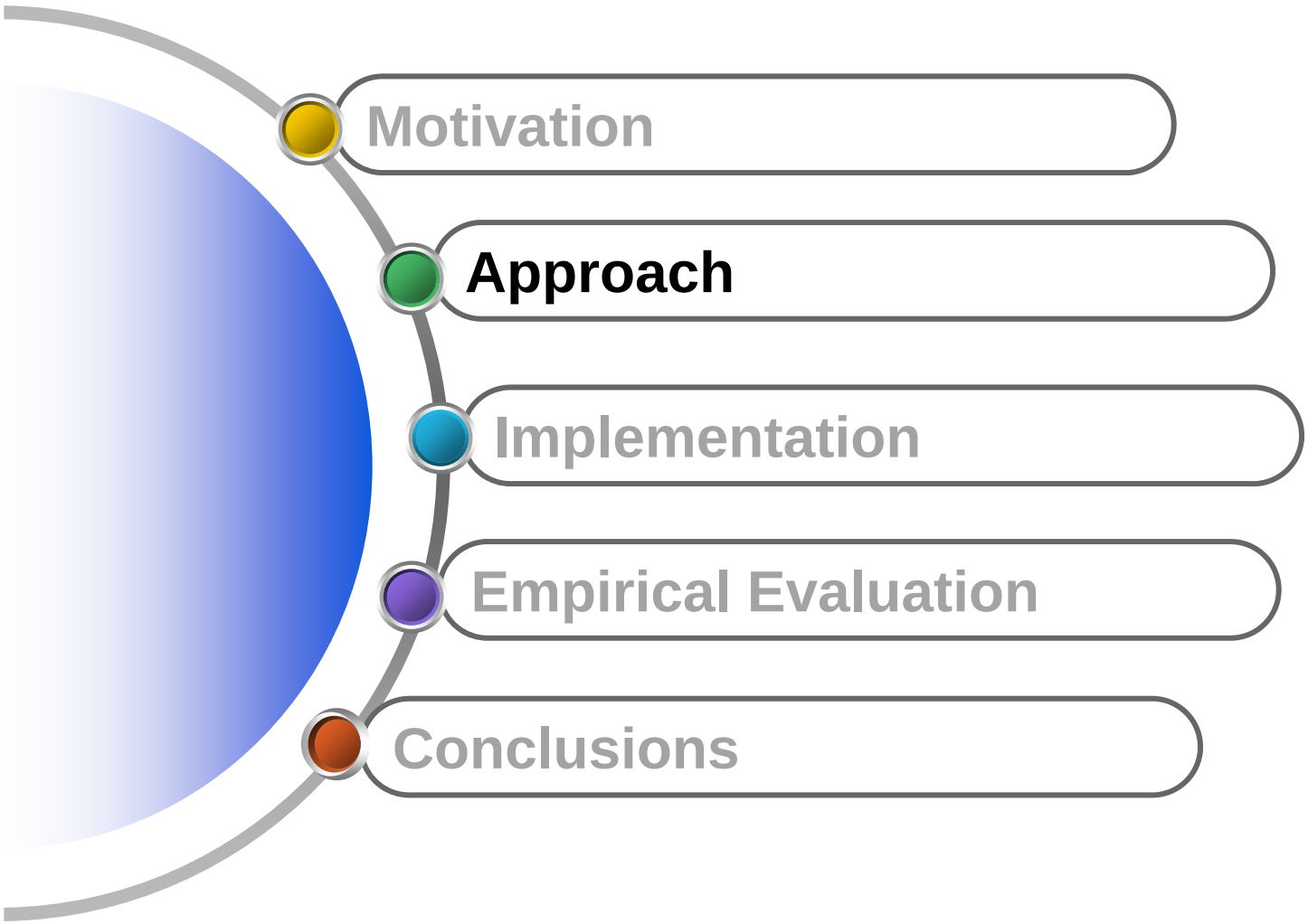
Problems		Single Version Testing	Regression Testing
Configuration Level	Selection	[11][15][18][25]	**Focus of this talk**
	Prioritization	NA	[18]
Test Case Level	Selection	[17]	[22][23][a]
	Prioritization	NA	[21] [a]

- Configuration **selection** [ICSM '12]
- Source code changes
- **Regression testing**
- Non-redundant
- Safe
- No test case selection

- Test case selection [ISSRE '11]
- Single version
- Configuration under test changes
- Non-redundant
- Safe

- Configuration prioritization [ISSTA '08]
- Source code changes
- Regression Testing
- No test case selection
- Redundant
- Not safe

Outline


Motivation
Approach
Implementation
Empirical Evaluation
Conclusions

Key Idea: Map configuration options to code

Key Idea: Map configuration options to code

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}

IE5

Key Idea: Map configuration options to code

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}

IE5

change

IE7

Key Idea: Map configuration options to code

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}

IE5

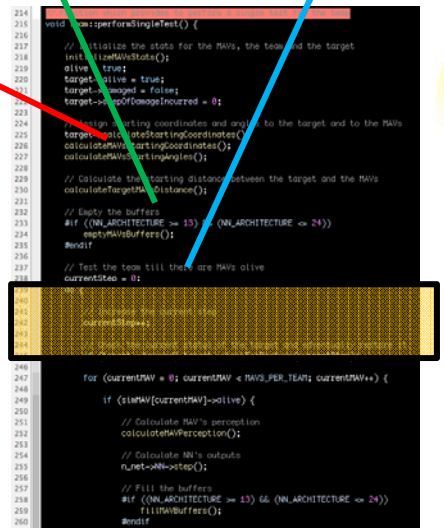
change

IE7

Key Idea: Map configuration options to code

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}

IE5



change

IE7

For ABB systems, configurable options (stored in a DB)
maps to variables in the source code

Key Idea: statically compute configuration option impact

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}

```
214  // Function which provides to perform a single test for the team
215  void Team::performSingleTest() {
216
217      // Initialize the stats for the MAVs, the team and the target
218      initializeMAVsStats();
219      alive = true;
220      target->alive = true;
221      target->damaged = false;
222      target->stepOfDamageIncurred = 0;
223
224      // Assign starting coordinates and angles to the target and to the MAVs
225      target->calculateStartingCoordinates();
226      calculateMAVsStartingCoordinates();
227      calculateMAVsStartingAngles();
228
229      // Calculate the starting distance between the target and the MAVs
230      calculateTargetMAVsDistance();
231
232      // Empty the buffers
233      #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234          emptyMAVsBuffers();
235      #endif
236
237      // Test the team till there are MAVs alive
238      currentStep = 0;
239      do {
240
241          // Increase the current step
242          currentStep++;
243
244          // Check the current status of the target and edventually restore it
245          if (target->damaged) target->checkForRestore(currentStep);
246
247          for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
248
249              if (simMAV[currentMAV]->alive) {
250
251                  // Calculate MAV's perception
252                  calculateMAVPerception();
253
254                  // Calculate NN's outputs
255                  n_net->NN->step();
256
257                  // Fill the buffers
258                  #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
259                      fillMAVBuffers();
260                  #endif
```

Key Idea: statically compute configuration option impact

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}

```
214  // Function which provides to perform a single test for the team
215  void Team::performSingleTest() {
216
217      // Initialize the stats for the MAVs, the team and the target
218      initializeMAVsStats();
219      alive = true;
220      target->alive = true;
221      target->damaged = false;
222      target->stepOfDamageIncurred = 0;
223
224      // Assign starting coordinates and angles to the target and to the MAVs
225      target->calculateStartingCoordinates();
226      calculateMAVsStartingCoordinates();
227      calculateMAVsStartingAngles();
228
229      // Calculate the starting distance between the target and the MAVs
230      calculateTargetMAVsDistance();
231
232      // Empty the buffers
233      #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234          emptyMAVsBuffers();
235      #endif
236
237      // Test the team till there are MAVs alive
238      currentStep = 0;
239      do {
240
241          // Increase the current step
242          currentStep++;
243
244          // Check the current status of the target and edventually restore it
245          if (target->damaged) target->checkForRestore(currentStep);
246
247          for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
248
249              if (simMAV[currentMAV]->alive) {
250
251                  // Calculate MAV's perception
252                  calculateMAVPerception();
253
254                  // Calculate NN's outputs
255                  n_net->NN->step();
256
257                  // Fill the buffers
258                  #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
259                      fillMAVBuffers();
260                  #endif
```

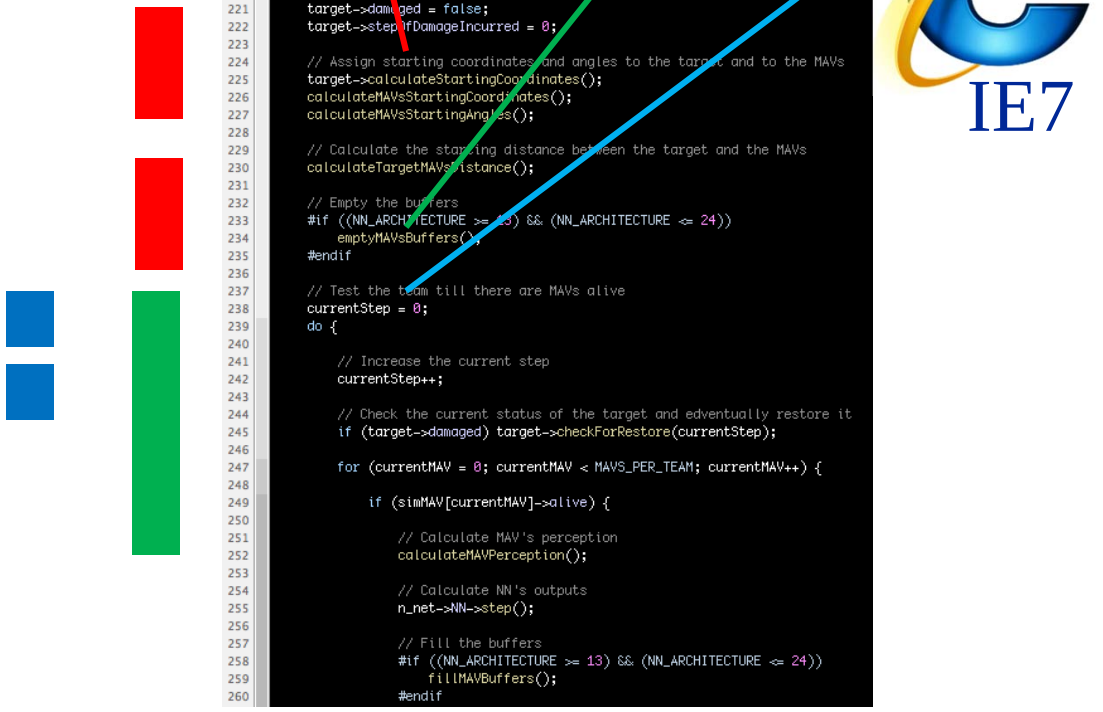
Key Idea: statically compute configuration option impact

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}

```
214  // Function which provides to perform a single test of the team
215  void Team::performSingleTest() {
216
217      // Initialize the stats for the MAVs, the team and the target
218      initializeMAVsStats();
219      alive = true;
220      target->alive = true;
221      target->damaged = false;
222      target->stepOfDamageIncurred = 0;
223
224      // Assign starting coordinates and angles to the target and to the MAVs
225      target->calculateStartingCoordinates();
226      calculateMAVsStartingCoordinates();
227      calculateMAVsStartingAngles();
228
229      // Calculate the starting distance between the target and the MAVs
230      calculateTargetMAVsDistance();
231
232      // Empty the buffers
233      #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234          emptyMAVsBuffers();
235      #endif
236
237      // Test the team till there are MAVs alive
238      currentStep = 0;
239      do {
240
241          // Increase the current step
242          currentStep++;
243
244          // Check the current status of the target and edventually restore it
245          if (target->damaged) target->checkForRestore(currentStep);
246
247          for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
248
249              if (simMAV[currentMAV]->alive) {
250
251                  // Calculate MAV's perception
252                  calculateMAVPerception();
253
254                  // Calculate NN's outputs
255                  n_net->NN->step();
256
257                  // Fill the buffers
258                  #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
259                      fillMAVBuffers();
260                  #endif
```

Key Idea: statically compute configuration option impact

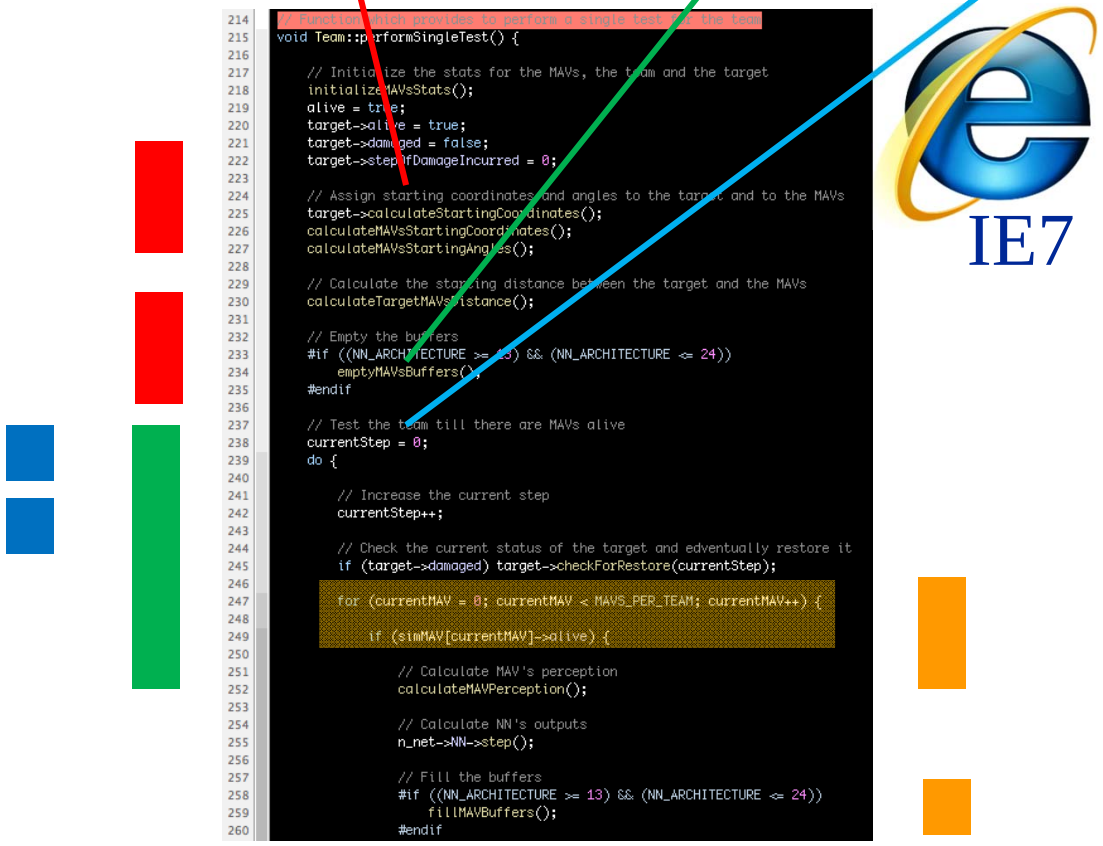
Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}



```
214  // Function which provides to perform a single test of the team
215  void Team::performSingleTest() {
216
217      // Initialize the stats for the MAVs, the team and the target
218      initializeMAVsStats();
219      alive = true;
220      target->alive = true;
221      target->damaged = false;
222      target->stepOfDamageIncurred = 0;
223
224      // Assign starting coordinates and angles to the target and to the MAVs
225      target->calculateStartingCoordinates();
226      calculateMAVsStartingCoordinates();
227      calculateMAVsStartingAngles();
228
229      // Calculate the starting distance between the target and the MAVs
230      calculateTargetMAVsDistance();
231
232      // Empty the buffers
233      #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234          emptyMAVsBuffers();
235      #endif
236
237      // Test the team till there are MAVs alive
238      currentStep = 0;
239      do {
240
241          // Increase the current step
242          currentStep++;
243
244          // Check the current status of the target and eventually restore it
245          if (target->damaged) target->checkForRestore(currentStep);
246
247          for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
248
249              if (simMAV[currentMAV]->alive) {
250
251                  // Calculate MAV's perception
252                  calculateMAVPerception();
253
254                  // Calculate NN's outputs
255                  n_net->NN->step();
256
257                  // Fill the buffers
258                  #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
259                      fillMAVBuffers();
260                  #endif
```

Key Idea: statically compute impact of the changes

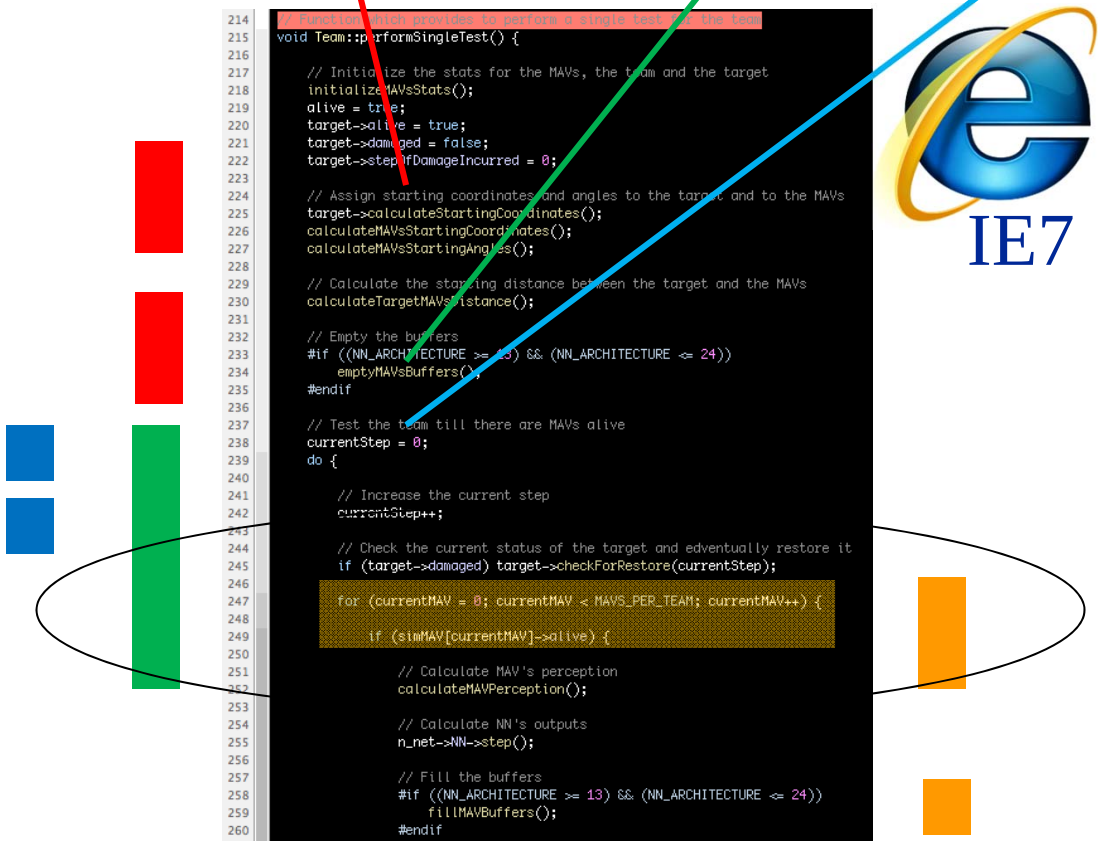
Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}




```
214  // Function which provides to perform a single test of the team
215  void Team::performSingleTest() {
216
217      // Initialize the stats for the MAVs, the team and the target
218      initializeMAVsStats();
219      alive = true;
220      target->alive = true;
221      target->damaged = false;
222      target->stepOfDamageIncurred = 0;
223
224      // Assign starting coordinates and angles to the target and to the MAVs
225      target->calculateStartingCoordinates();
226      calculateMAVsStartingCoordinates();
227      calculateMAVsStartingAngles();
228
229      // Calculate the starting distance between the target and the MAVs
230      calculateTargetMAVsDistance();
231
232      // Empty the buffers
233      #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234          emptyMAVsBuffers();
235      #endif
236
237      // Test the team till there are MAVs alive
238      currentStep = 0;
239      do {
240
241          // Increase the current step
242          currentStep++;
243
244          // Check the current status of the target and edventually restore it
245          if (target->damaged) target->checkForRestore(currentStep);
246
247          for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
248
249              if (simMAV[currentMAV]->alive) {
250
251                  // Calculate MAV's perception
252                  calculateMAVPerception();
253
254                  // Calculate NN's outputs
255                  n_net->NN->step();
256
257                  // Fill the buffers
258                  #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
259                      fillMAVBuffers();
260                  #endif
```

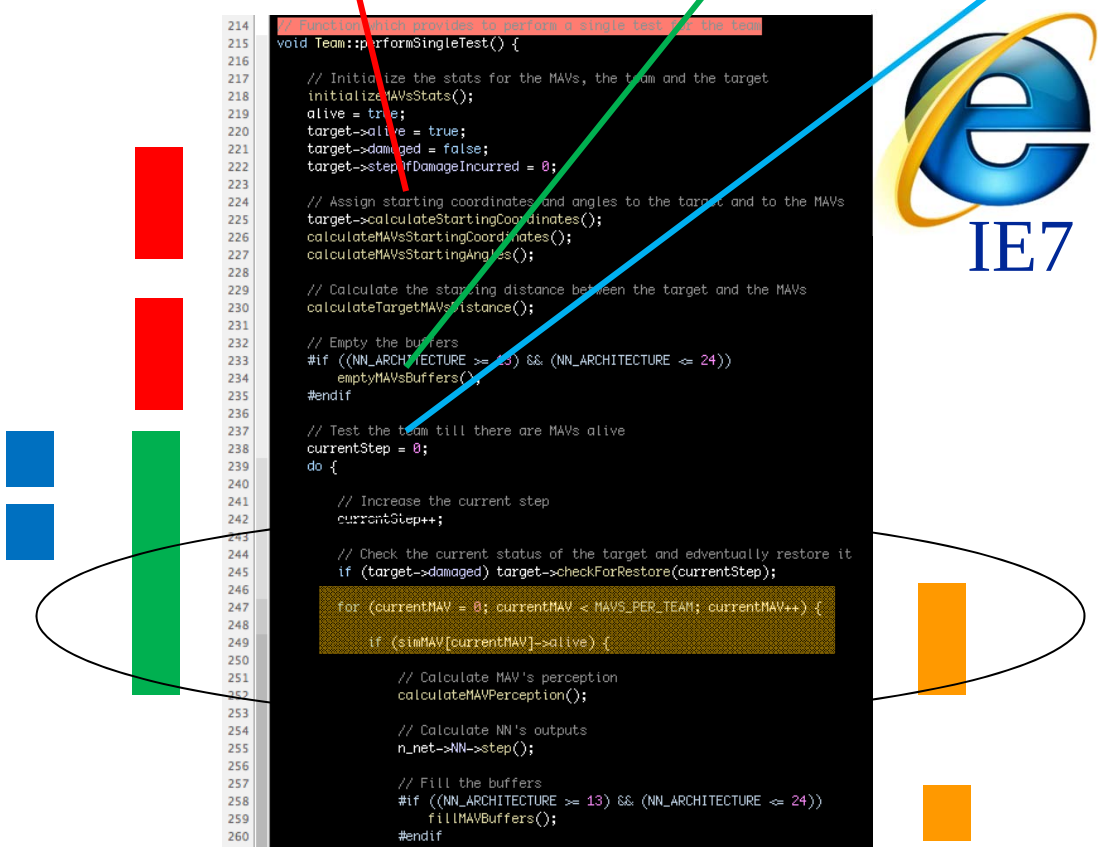
Key Idea: Intersect configuration impact with change impact

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}



Key Idea: Intersect configuration impact with change impact

Configuration options: {pop-up-blocker, Google Toolbar, Do Not Track}



Select configuration option “Google Toolbar” for regression testing

Safely discard “pop-up blocker” and “Do Not Track”

Example Program
Mapping configurable options to source code

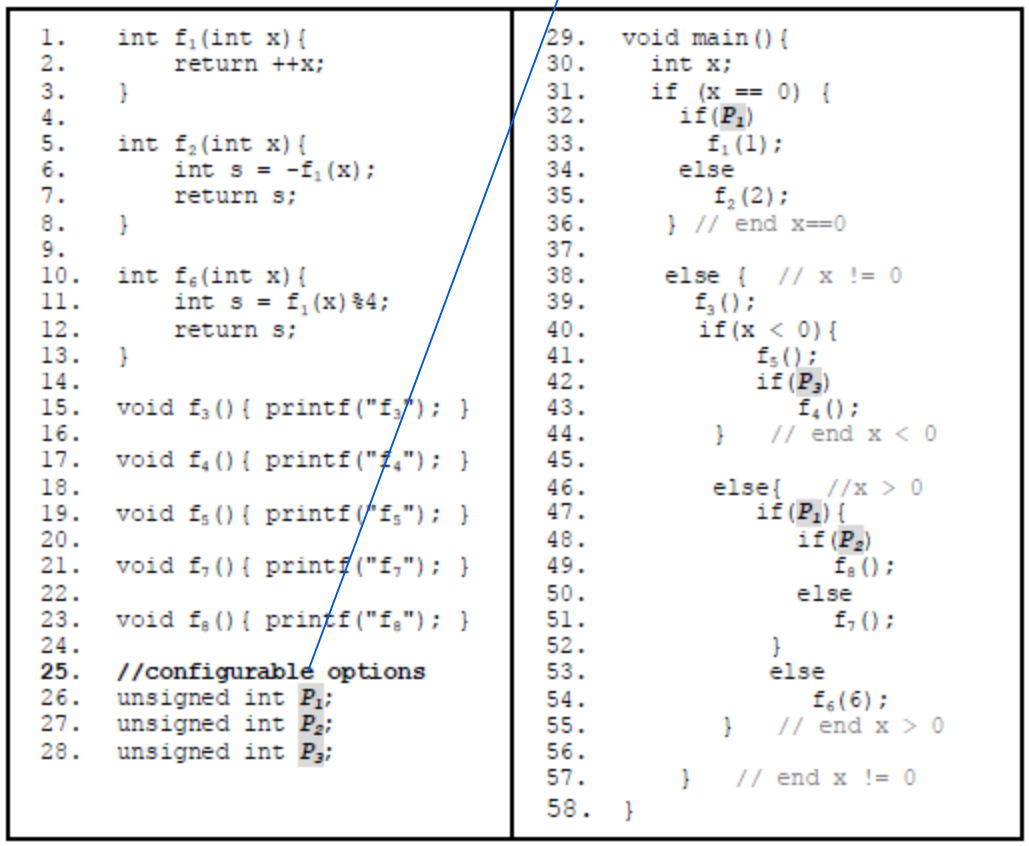
Configurable Options: {P_1, P_2, P_3}

```
1.   int f1(int x){
2.       return ++x;
3.   }
4.
5.   int f2(int x){
6.       int s = -f1(x);
7.       return s;
8.   }
9.
10.  int f6(int x){
11.      int s = f1(x)%4;
12.      return s;
13.  }
14.
15.  void f3(){ printf("f3"); }
16.
17.  void f4(){ printf("f4"); }
18.
19.  void f5(){ printf("f5"); }
20.
21.  void f7(){ printf("f7"); }
22.
23.  void f8(){ printf("f8"); }
24.
25.  //configurable options
26.  unsigned int P1;
27.  unsigned int P2;
28.  unsigned int P3;
```

```
29.  void main(){
30.    int x;
31.    if (x == 0) {
32.      if(P1)
33.        f1(1);
34.      else
35.        f2(2);
36.    } // end x==0
37.
38.    else {  // x != 0
39.      f3();
40.      if(x < 0){
41.          f5();
42.          if(P3)
43.              f4();
44.       }   // end x < 0
45.
46.       else{   //x > 0
47.          if(P1){
48.              if(P2)
49.                  f8();
50.              else
51.                  f7();
52.          }
53.          else
54.              f6(6);
55.       }   // end x > 0
56.
57.    }   // end x != 0
58.  }
```

Example Program
Mapping configurable options to source code

Configurable Options: {P_1, P_2, P_3}

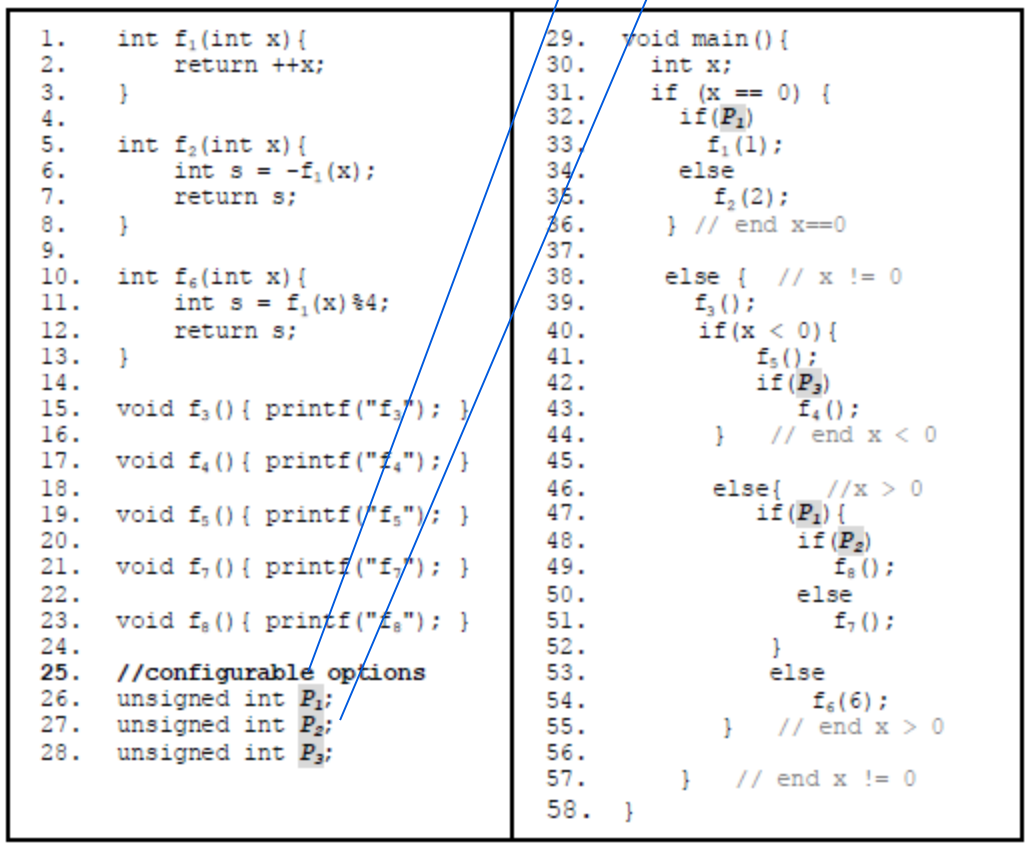


```
1.   int f1(int x){
2.       return ++x;
3.   }
4.
5.   int f2(int x){
6.       int s = -f1(x);
7.       return s;
8.   }
9.
10.  int f6(int x){
11.      int s = f1(x)%4;
12.      return s;
13.  }
14.
15.  void f3(){ printf("f3"); }
16.
17.  void f4(){ printf("f4"); }
18.
19.  void f5(){ printf("f5"); }
20.
21.  void f7(){ printf("f7"); }
22.
23.  void f8(){ printf("f8"); }
24.
25.  //configurable options
26.  unsigned int P1;
27.  unsigned int P2;
28.  unsigned int P3;
```

```
29.  void main(){
30.    int x;
31.    if (x == 0) {
32.      if(P1)
33.        f1(1);
34.      else
35.        f2(2);
36.    } // end x==0
37.
38.    else {  // x != 0
39.      f3();
40.      if(x < 0){
41.          f5();
42.          if(P3)
43.              f4();
44.       }   // end x < 0
45.
46.       else{   //x > 0
47.          if(P1){
48.              if(P2)
49.                  f8();
50.              else
51.                  f7();
52.          }
53.          else
54.              f6(6);
55.       }   // end x > 0
56.
57.    }   // end x != 0
58.  }
```

Example Program
Mapping configurable options to source code

Configurable Options: {P_1, P_2, P_3}

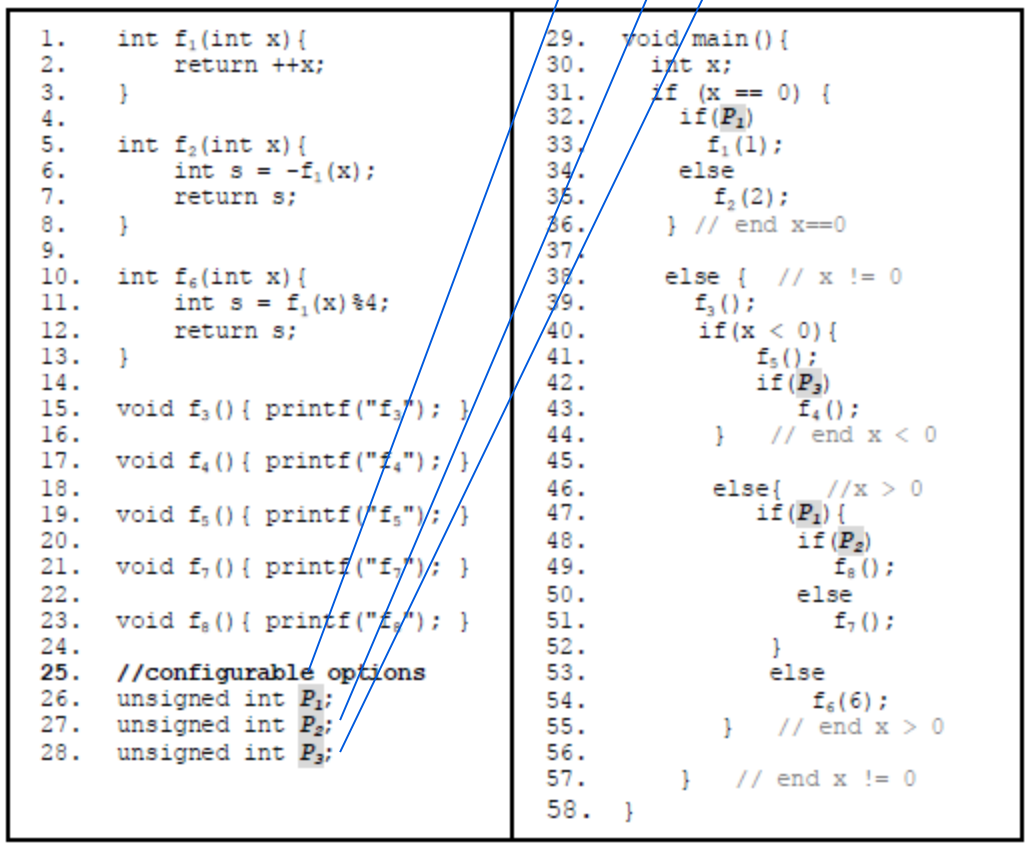


```
1.   int f1(int x){
2.       return ++x;
3.   }
4.
5.   int f2(int x){
6.       int s = -f1(x);
7.       return s;
8.   }
9.
10.  int f6(int x){
11.      int s = f1(x)%4;
12.      return s;
13.  }
14.
15.  void f3(){ printf("f3"); }
16.
17.  void f4(){ printf("f4"); }
18.
19.  void f5(){ printf("f5"); }
20.
21.  void f7(){ printf("f7"); }
22.
23.  void f8(){ printf("f8"); }
24.
25.  //configurable options
26.  unsigned int P1;
27.  unsigned int P2;
28.  unsigned int P3;
```

```
29.  void main(){
30.    int x;
31.    if (x == 0) {
32.      if(P1)
33.        f1(1);
34.      else
35.        f2(2);
36.    } // end x==0
37.
38.    else {  // x != 0
39.      f3();
40.      if(x < 0){
41.          f5();
42.          if(P3)
43.              f4();
44.      }   // end x < 0
45.
46.      else{   //x > 0
47.          if(P1){
48.              if(P2)
49.                  f8();
50.              else
51.                  f7();
52.          }
53.          else
54.              f6(6);
55.      }   // end x > 0
56.
57.    }   // end x != 0
58.  }
```

Example Program
Mapping configurable options to source code

Configurable Options: {P_1, P_2, P_3}



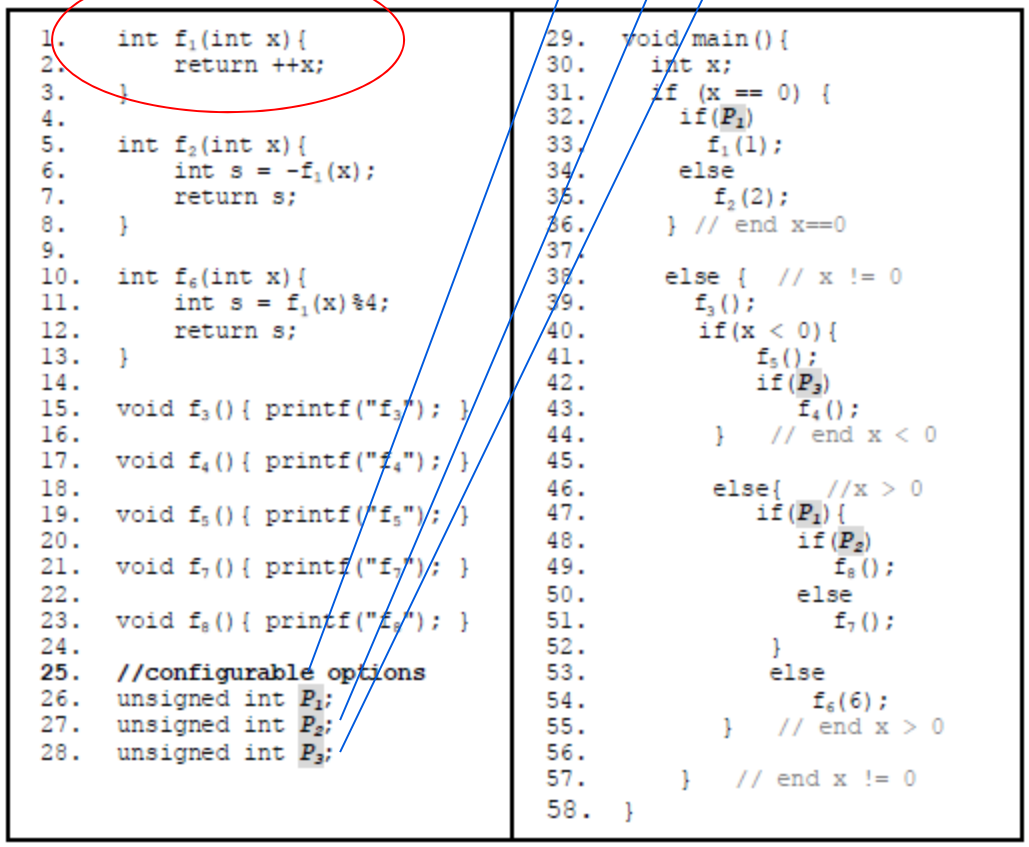
```
1.   int f1(int x){
2.       return ++x;
3.   }
4.
5.   int f2(int x){
6.       int s = -f1(x);
7.       return s;
8.   }
9.
10.  int f6(int x){
11.      int s = f1(x)%4;
12.      return s;
13.  }
14.
15.  void f3(){ printf("f3"); }
16.
17.  void f4(){ printf("f4"); }
18.
19.  void f5(){ printf("f5"); }
20.
21.  void f7(){ printf("f7"); }
22.
23.  void f8(){ printf("f8"); }
24.
25.  //configurable options
26.  unsigned int P1;
27.  unsigned int P2;
28.  unsigned int P3;
```

```
29.  void main(){
30.    int x;
31.    if (x == 0) {
32.      if(P1)
33.        f1(1);
34.      else
35.        f2(2);
36.    } // end x==0
37.
38.    else {  // x != 0
39.      f3();
40.      if(x < 0){
41.          f5();
42.          if(P3)
43.              f4();
44.      }   // end x < 0
45.
46.      else{   //x > 0
47.          if(P1){
48.              if(P2)
49.                  f8();
50.              else
51.                  f7();
52.          }
53.          else
54.              f6(6);
55.      }   // end x > 0
56.
57.    }   // end x != 0
58.  }
```

Example Program
Mapping configurable options to source code

Configurable Options: {P_1, P_2, P_3}

Function f_1 changes



```
1.   int f1(int x){
2.       return ++x;
3.   }
4.
5.   int f2(int x){
6.       int s = -f1(x);
7.       return s;
8.   }
9.
10.  int f6(int x){
11.      int s = f1(x)%4;
12.      return s;
13.  }
14.
15.  void f3(){ printf("f3"); }
16.
17.  void f4(){ printf("f4"); }
18.
19.  void f5(){ printf("f5"); }
20.
21.  void f7(){ printf("f7"); }
22.
23.  void f8(){ printf("f8"); }
24.
25.  //configurable options
26.  unsigned int P1;
27.  unsigned int P2;
28.  unsigned int P3;
```

```
29.  void main(){
30.    int x;
31.    if (x == 0) {
32.      if(P1)
33.        f1(1);
34.      else
35.        f2(2);
36.    } // end x==0
37.
38.    else {  // x != 0
39.      f3();
40.      if(x < 0){
41.          f5();
42.          if(P3)
43.              f4();
44.      }   // end x < 0
45.
46.      else{   //x > 0
47.          if(P1){
48.              if(P2)
49.                  f8();
50.              else
51.                  f7();
52.          }
53.          else
54.              f6(6);
55.      }   // end x > 0
56.
57.    }   // end x != 0
58.  }
```

Example

Example

Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Example

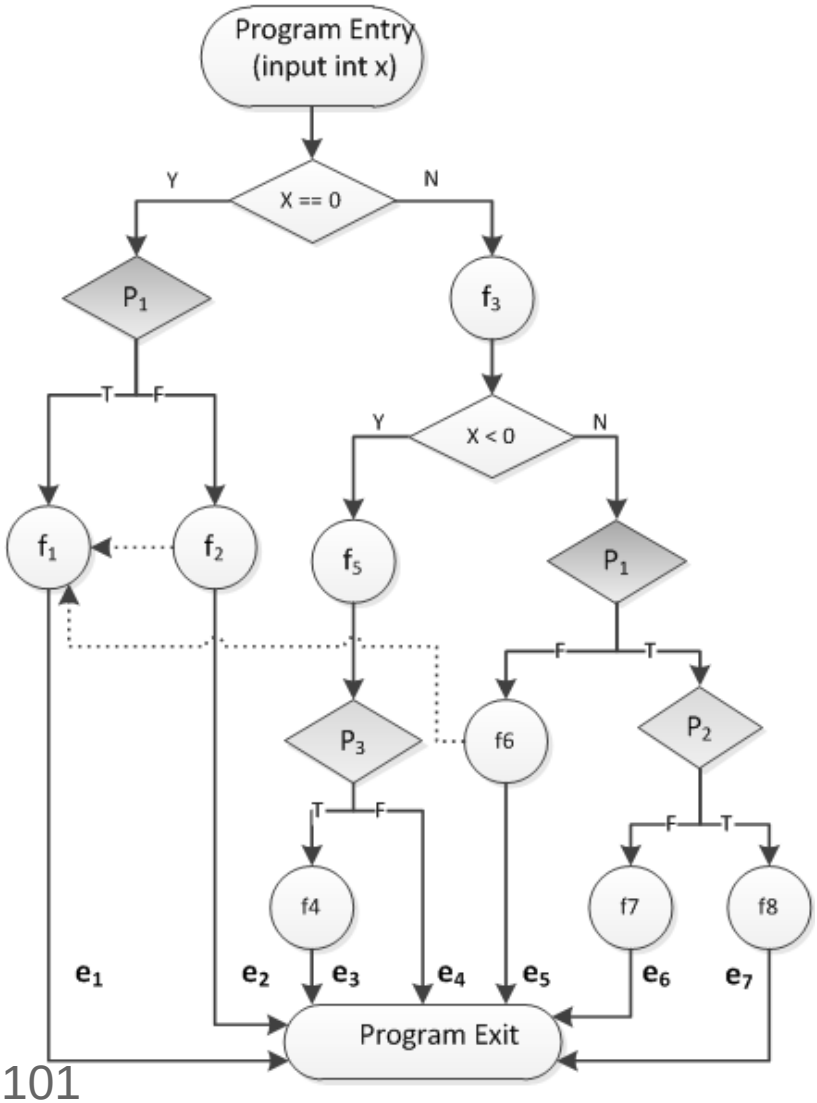
Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Configurations by pair-wise CIT

	P_1	P_2	P_3
C_1	True	True	True
C_2	True	False	False
C_3	False	True	False
C_4	False	False	True

Example
Simplified dependency graph



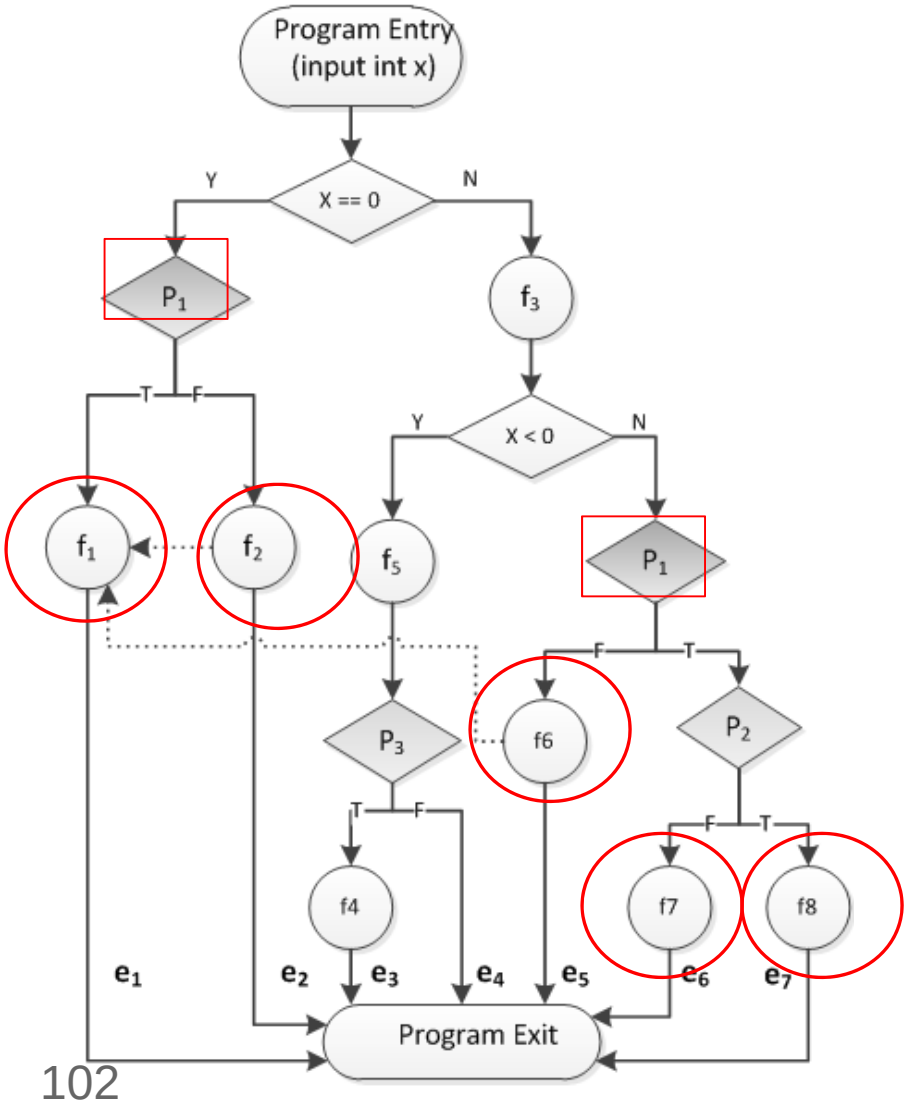
Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Configurations by pair-wise CIT

	P_1	P_2	P_3
C_1	True	True	True
C_2	True	False	False
C_3	False	True	False
C_4	False	False	True

Impact of configuration option P_1
f_1, f_2, f_6, f_7, and f_8



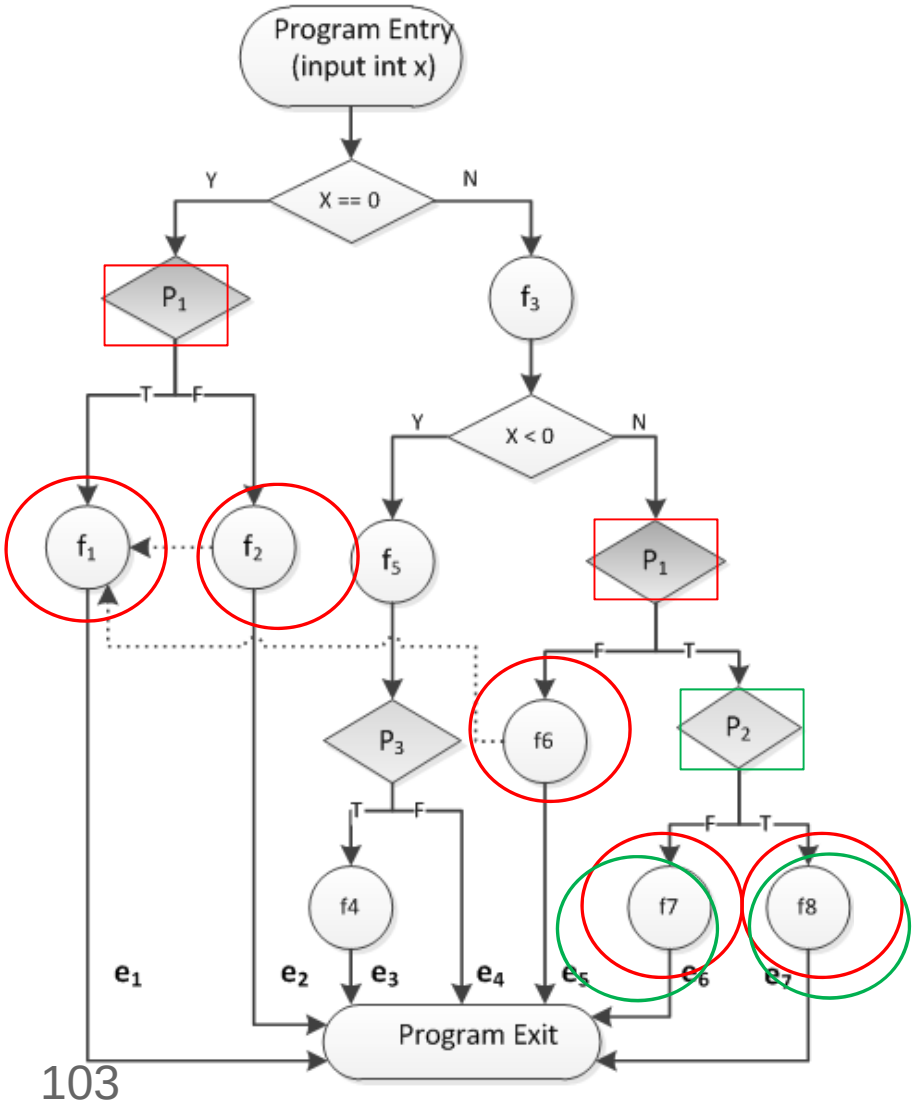
Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Configurations by pair-wise CIT

	P_1	P_2	P_3
C_1	True	True	True
C_2	True	False	False
C_3	False	True	False
C_4	False	False	True

Impact of configuration option P_2
f_7 and f_8



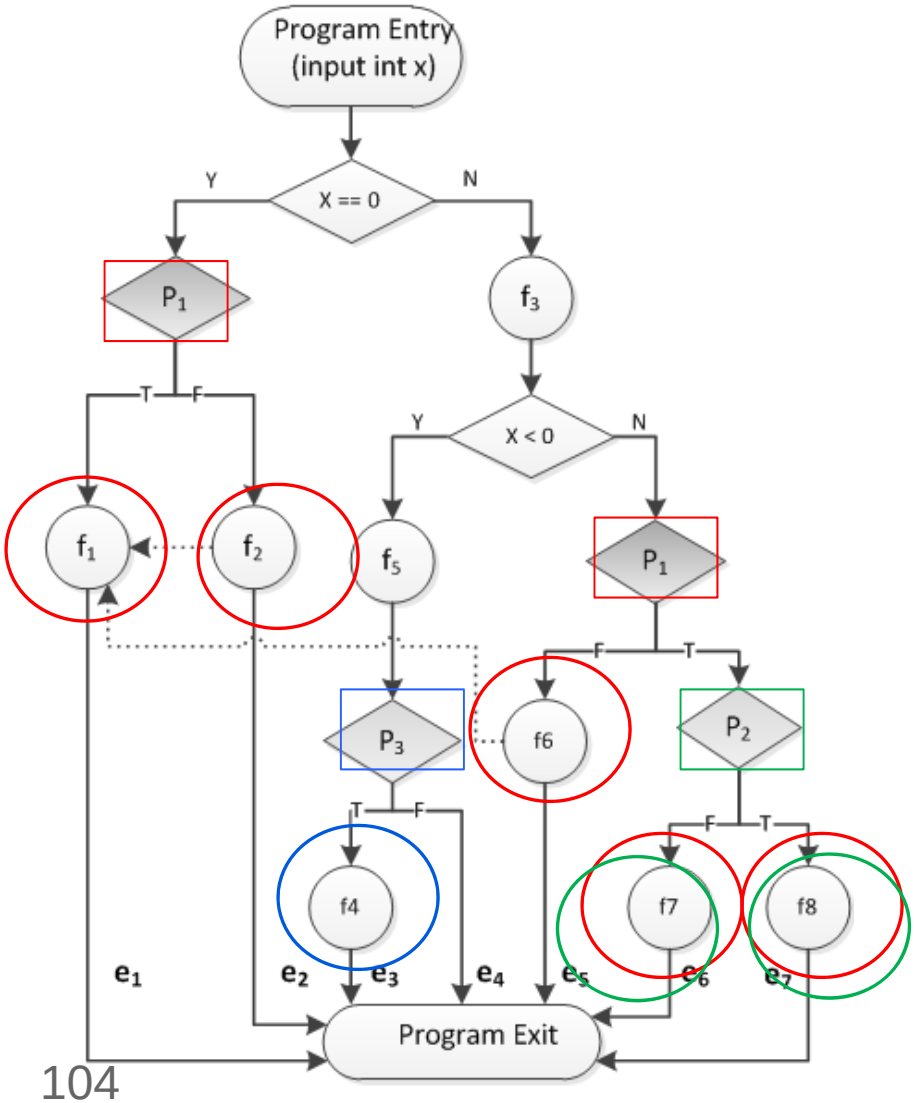
Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Configurations by pair-wise CIT

	P_1	P_2	P_3
C_1	True	True	True
C_2	True	False	False
C_3	False	True	False
C_4	False	False	True

Impact of configuration option P_3 f_4



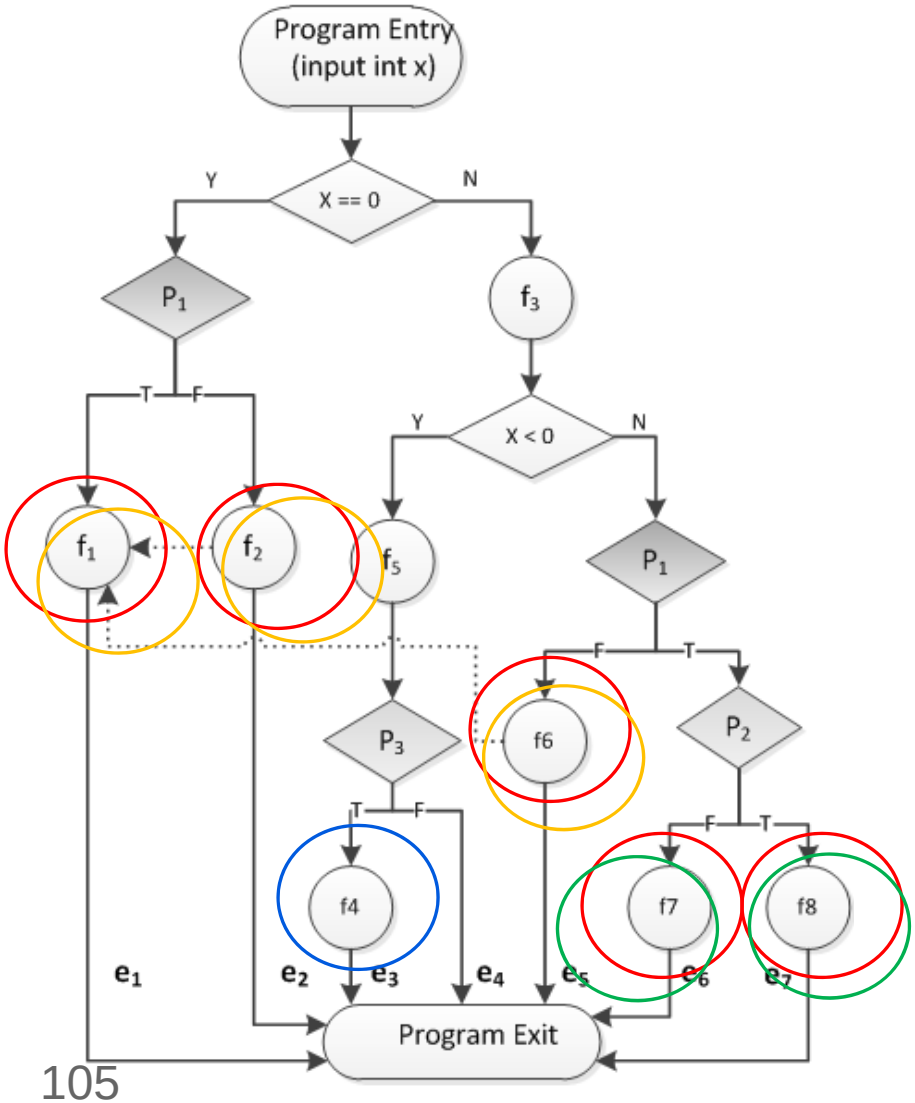
Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Configurations by pair-wise CIT

	P_1	P_2	P_3
C_1	True	True	True
C_2	True	False	False
C_3	False	True	False
C_4	False	False	True

Impact of changed function f_1
f_1, f_2, and f_6



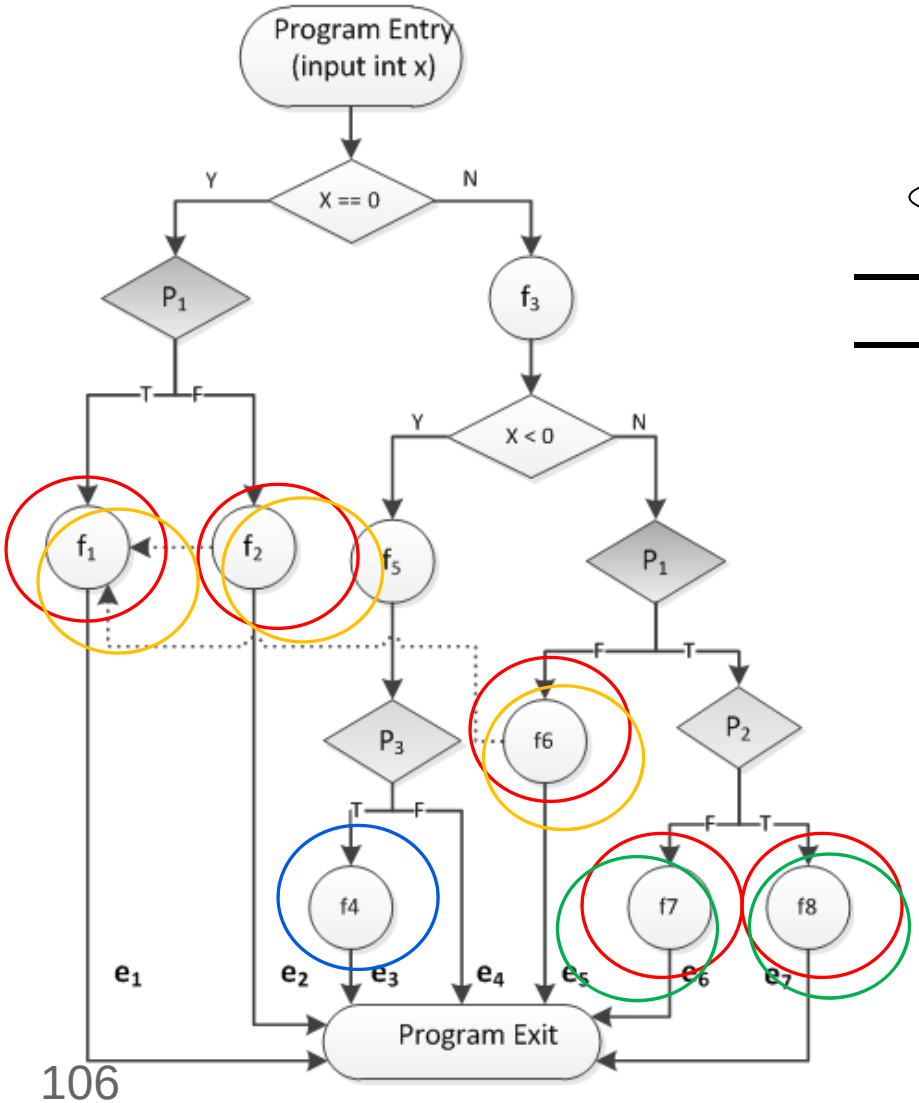
Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Configurations by pair-wise CIT

	P_1	P_2	P_3
C_1	True	True	True
C_2	True	False	False
C_3	False	True	False
C_4	False	False	True

Select option P_1 and safely discard P_2 and P_3



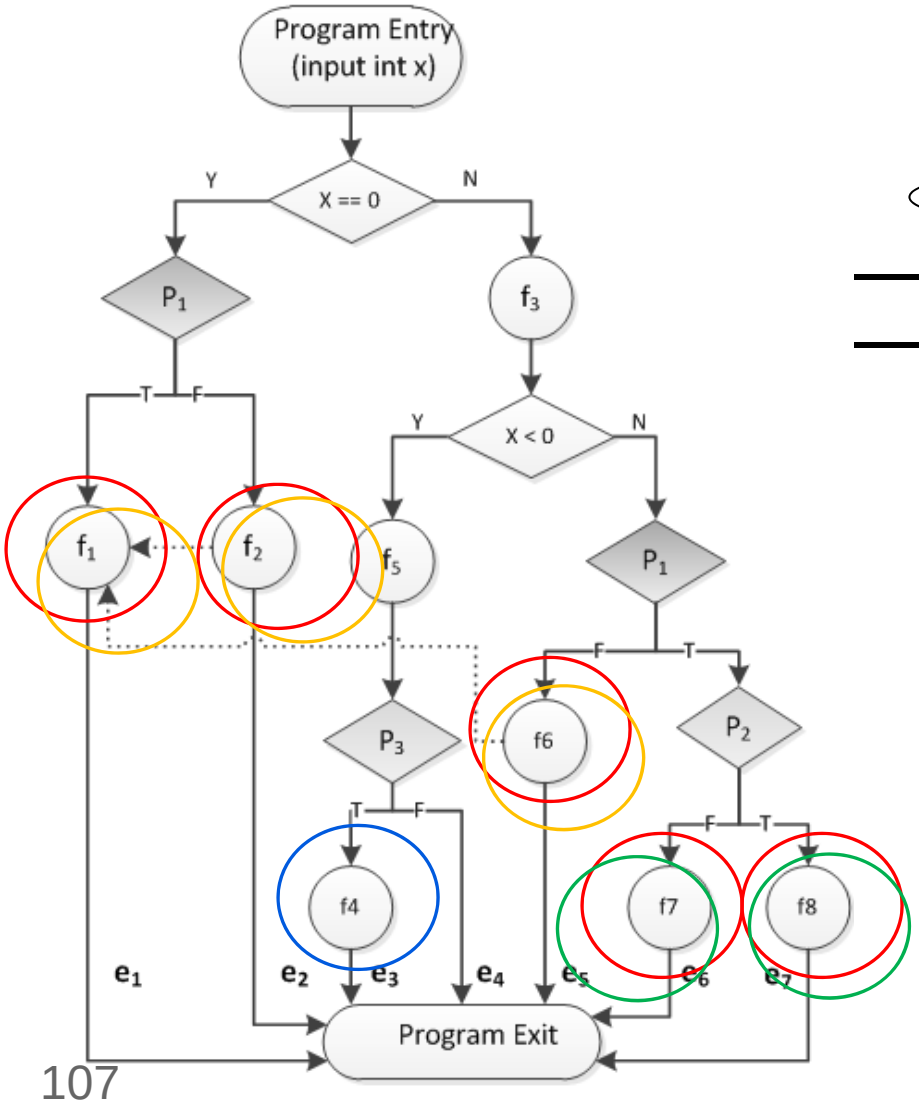
Configurable Options

Options	Values	
P_1	True	False
~~P_2~~	~~True~~	~~False~~
~~P_3~~	~~True~~	~~False~~

Configurations by pair-wise CIT

	P_1	P_2	P_3
C_1	True	True	True
C_2	True	False	False
C_3	False	True	False
C_4	False	False	True

Select option P_1 and safely discard P_2 and P_3



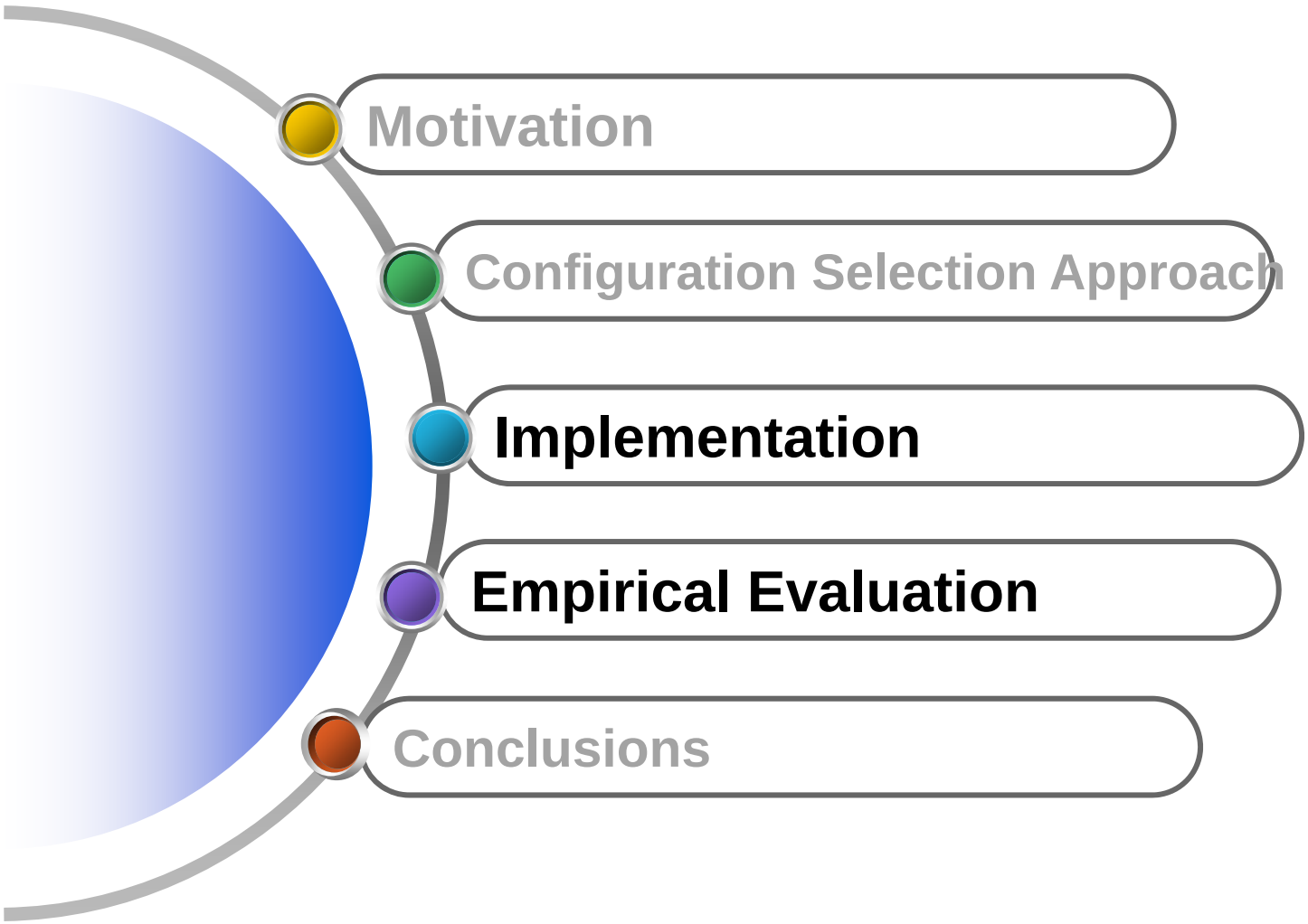
Configurable Options

Options	Values	
P_1	True	False
~~P_2~~	~~True~~	~~False~~
~~P_3~~	~~True~~	~~False~~

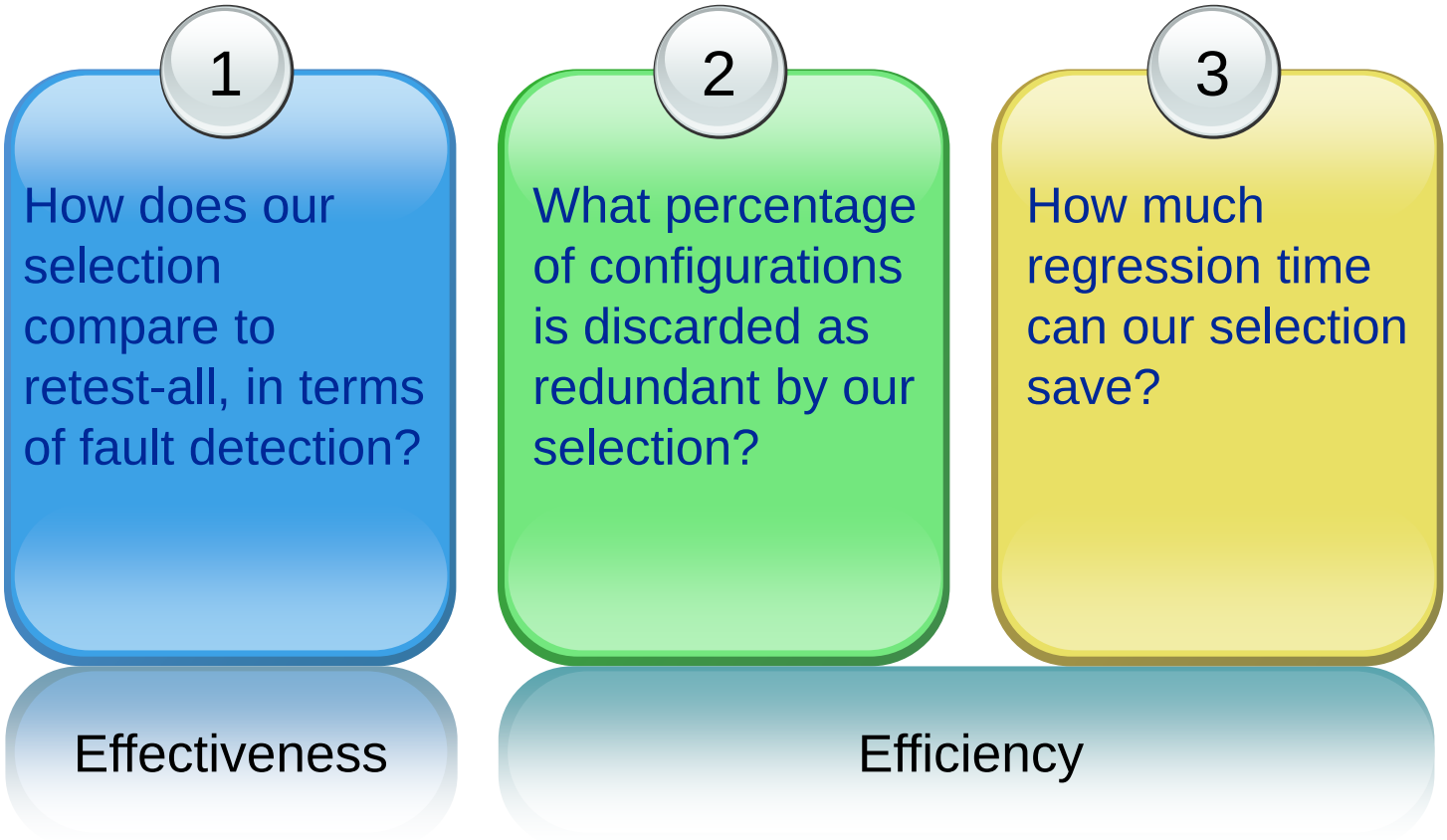
Configurations by pair-wise CIT

	P_1	P_2	P_3
C_1	True	True	True
~~C_2~~	~~True~~	~~False~~	~~False~~
C_3	False	True	False
~~C_4~~	~~False~~	~~False~~	~~True~~

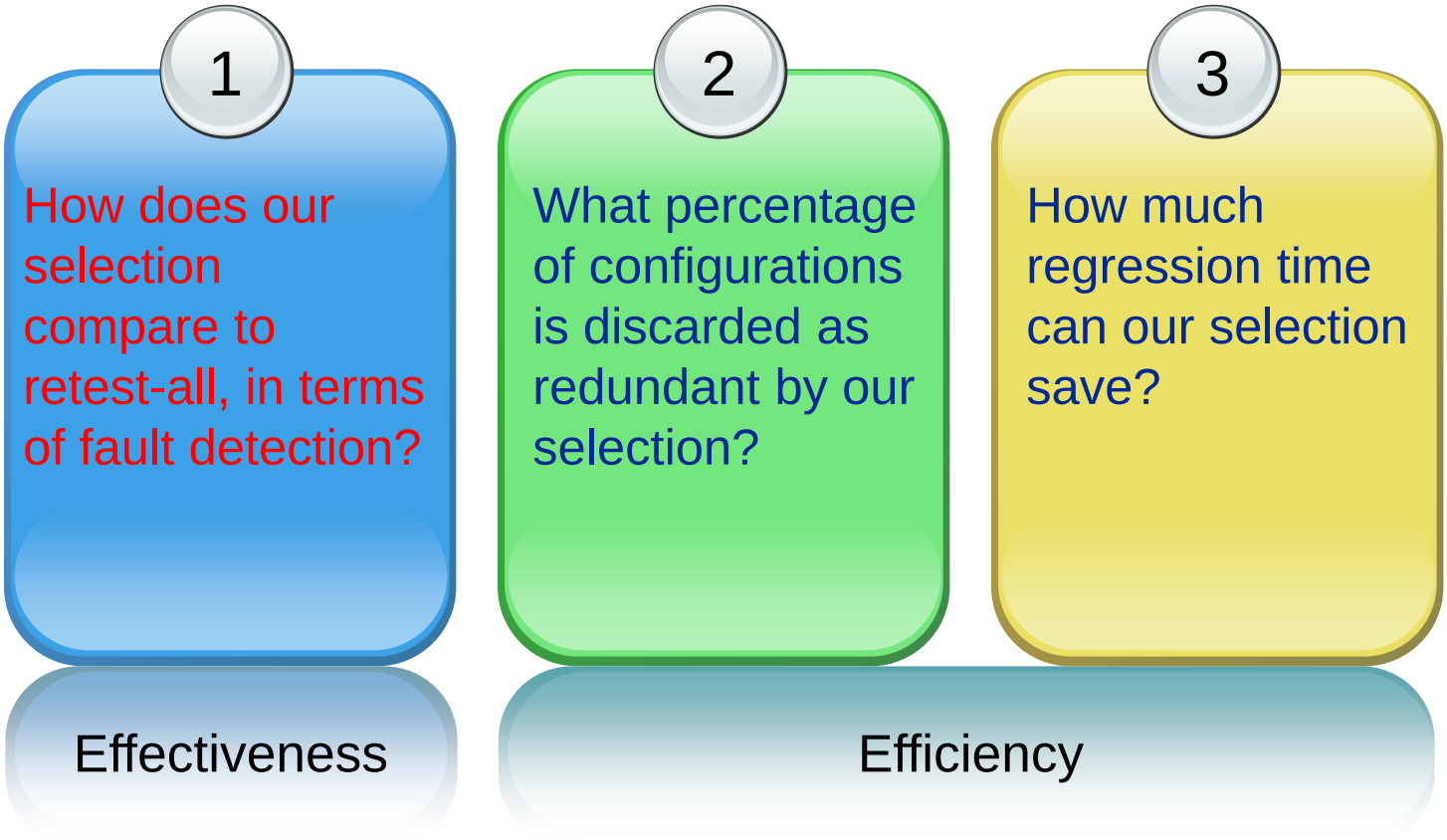
Outline


Motivation
Configuration Selection Approach
Implementation
Empirical Evaluation
Conclusions

Research Questions



Research Questions



Subjects

- *Make* (Software Infrastructure Repository)
  - *V3.77 to v3.78.1*
  - *LOC: ≈ 15k LOC*
  - *Code changes:* selects 60 from 869
  - *Seeded 15 faults*
  - *Configurable options*: 11 (binary) → 7 configurations
- *Grep*
  - *V1.0 to V2.0*
  - *LOC: ≈ 8k LOC*
  - *Code changes: 15*
  - *Seeded 15 faults*
  - *Configurable options:* 14 (binary) → 7 configurations

Results

Fault Detection Ability

	Make	*Grep*
Retest-all	8/15	6/15
Our selection	8/15	6/15
Random selection	3/15	5/15

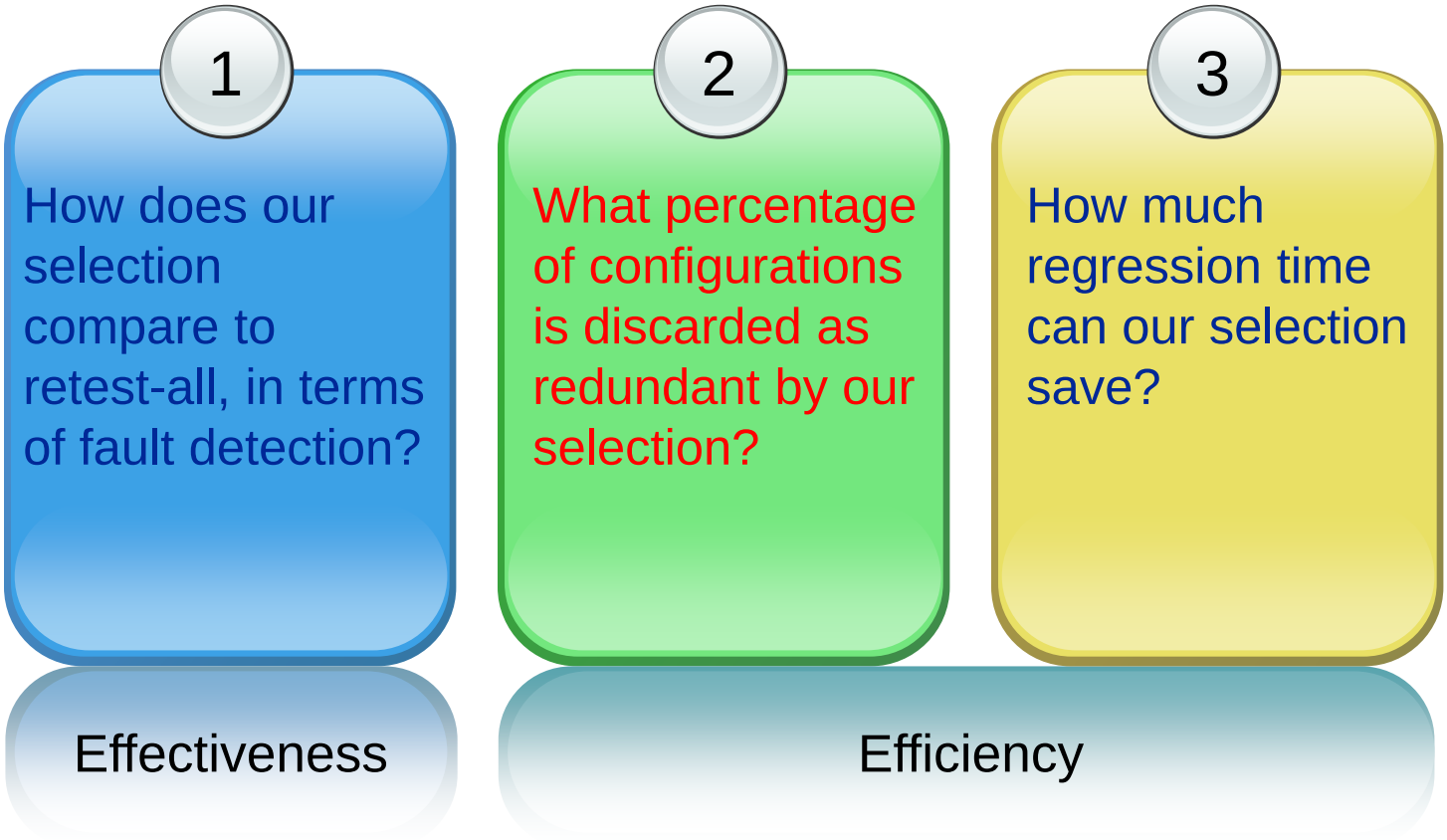
Results

Fault Detection Ability

	Make	*Grep*
Retest-all	8/15	6/15
Our selection	8/15	6/15
Random selection	3/15	5/15

Our approach is safe wrt retest-all configurations

Research Questions



Subject
ABB1

- *LOC*: 1.18 MLOC
- *Number of Functions:* 20,432 functions
- *Code changes:* 203
- Configurable options: 545 (number of values range from 2 to 9) → 159 configurations
- Among the 203 changes, we selected three sets of 30 changes for analysis

Results
Percentage of configurations selected

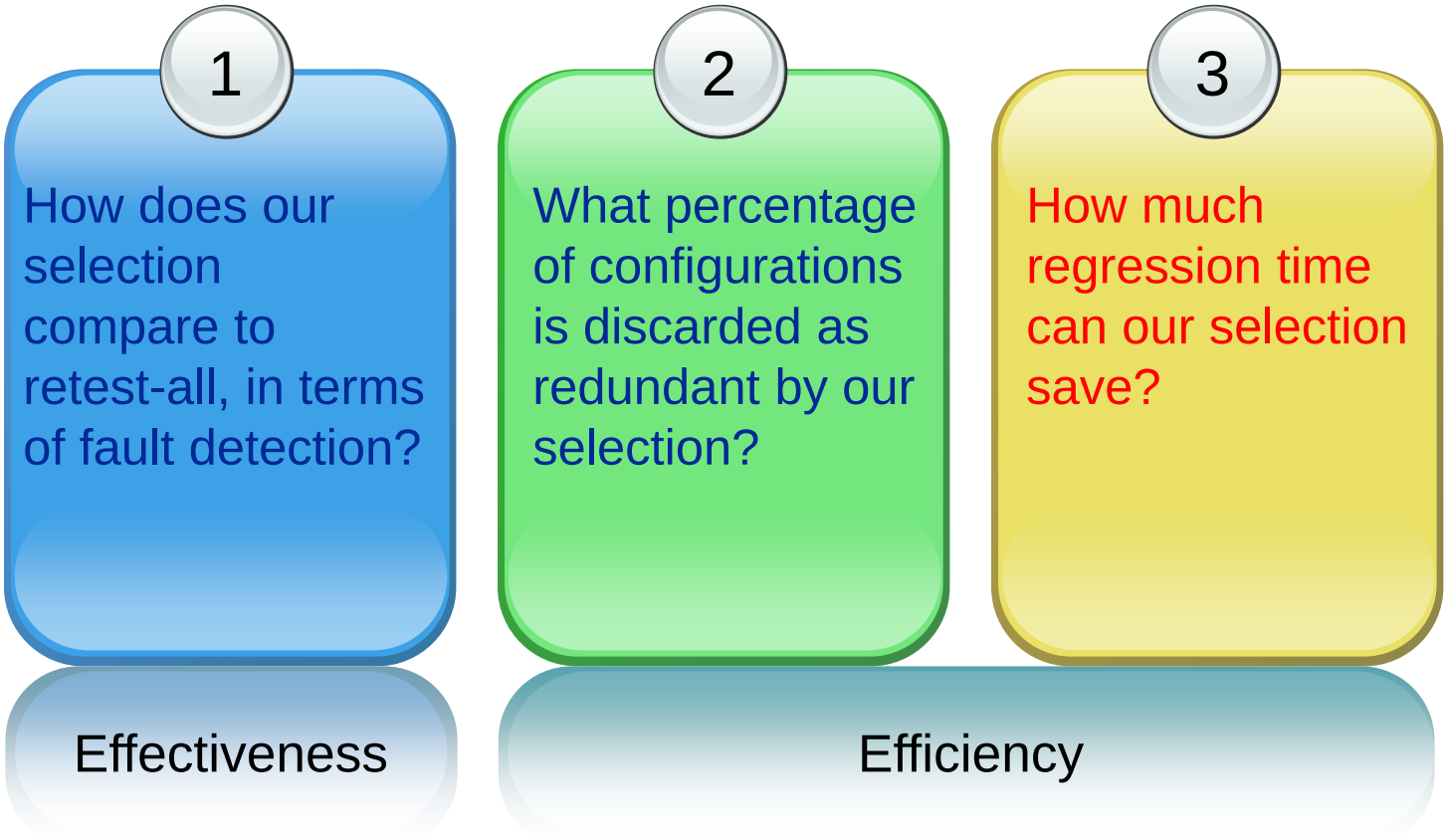
NUMBER OF CONFIGURABLE OPTIONS SELECTED

	Change set 1	Change set 2	Change set 3	Average
Retest-all	545			
Selected	167	161	161	163
reduction	69%	70%	70%	70%

NUMBER OF CONFIGURATIONS SELECTED

	Change set 1	Change set 2	Change set 3	Average
Retest-all	159			
Selected	120	120	120	120
reduction	25%	25%	25%	25%

Research Questions



Results
Testing time savings

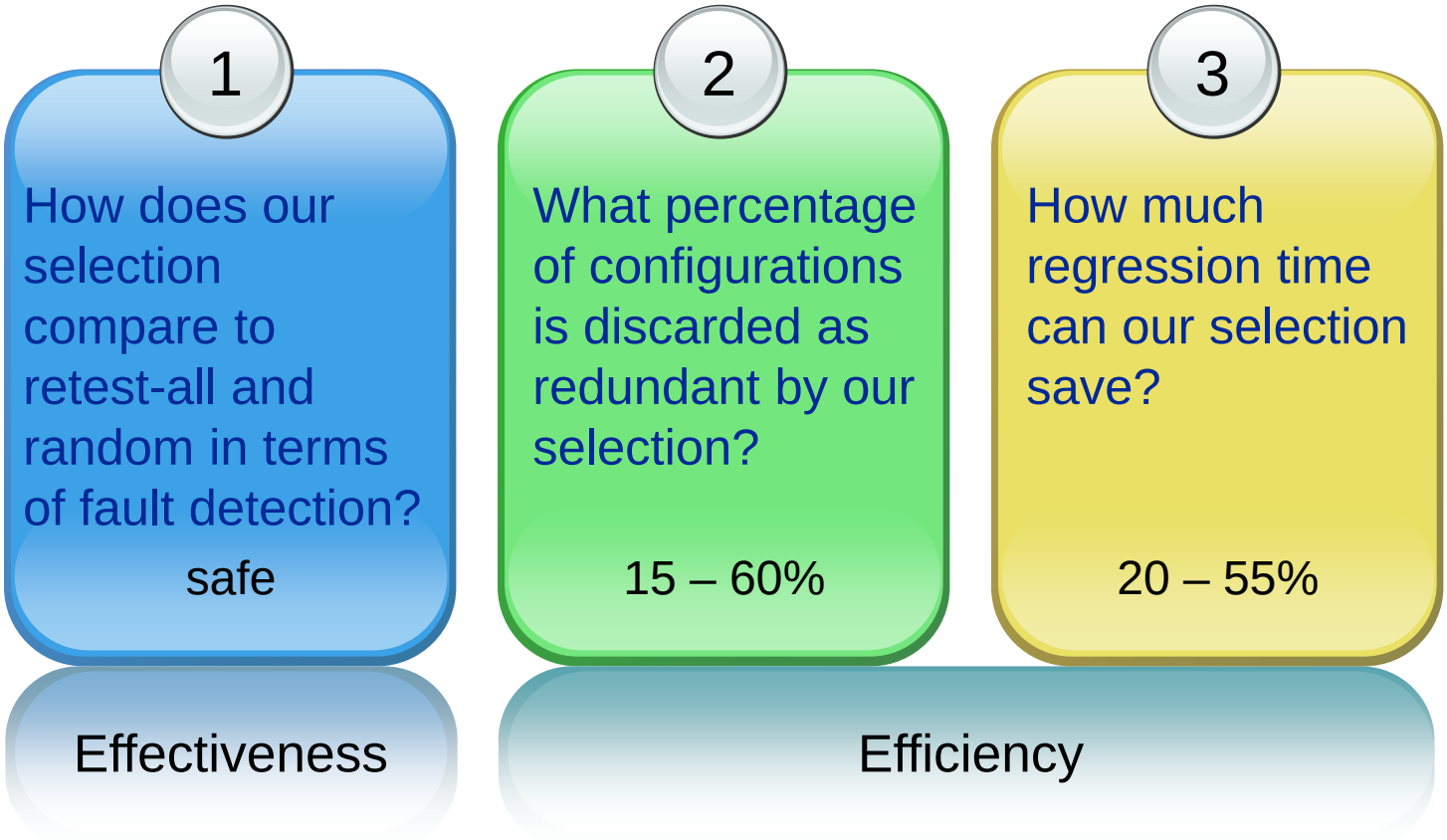
		grep	make	ABB1
Testing time	Retest-all	70m	700m	795h
	Our approach	60m	300m	600h
Overhead of selection		5.2m	13m	28h
Time savings		5m	387m	167h
		50%	**55%**	**21%**

Results
Testing time savings

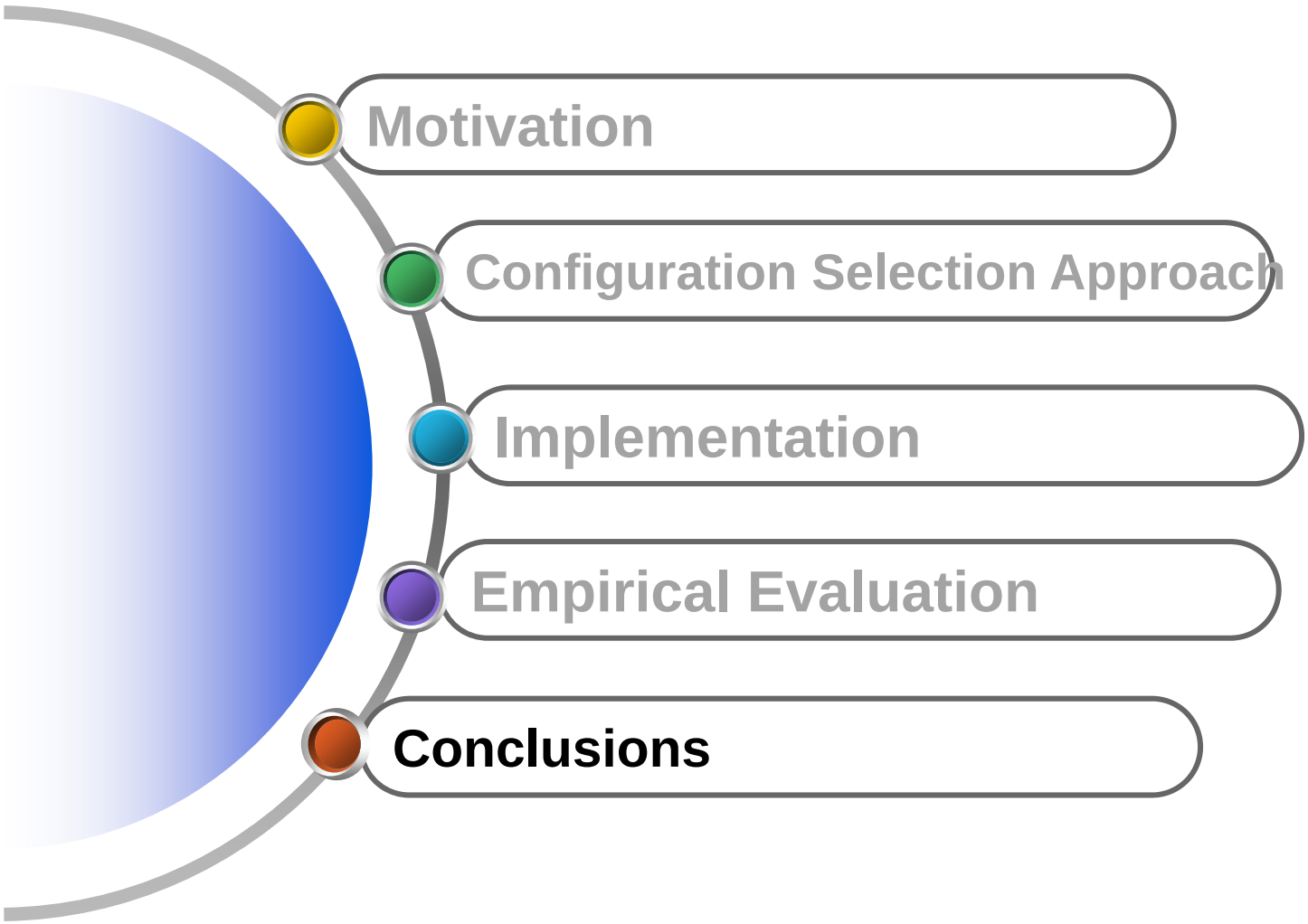
		grep	make	ABB1
Testing time	Retest-all	70m	700m	795h
	Our approach	60m	300m	600h
Overhead of selection		5.2m	13m	28h
Time savings		5m	387m	167h
		50%	**55%**	**21%**

Our configuration selection approach saves about 20-55% of testing time wrt retest-all configurations

Better than random, safe wrt retest-all



Outline


Motivation
Configuration Selection Approach
Implementation
Empirical Evaluation
Conclusions

First configuration selection approach for regression testing

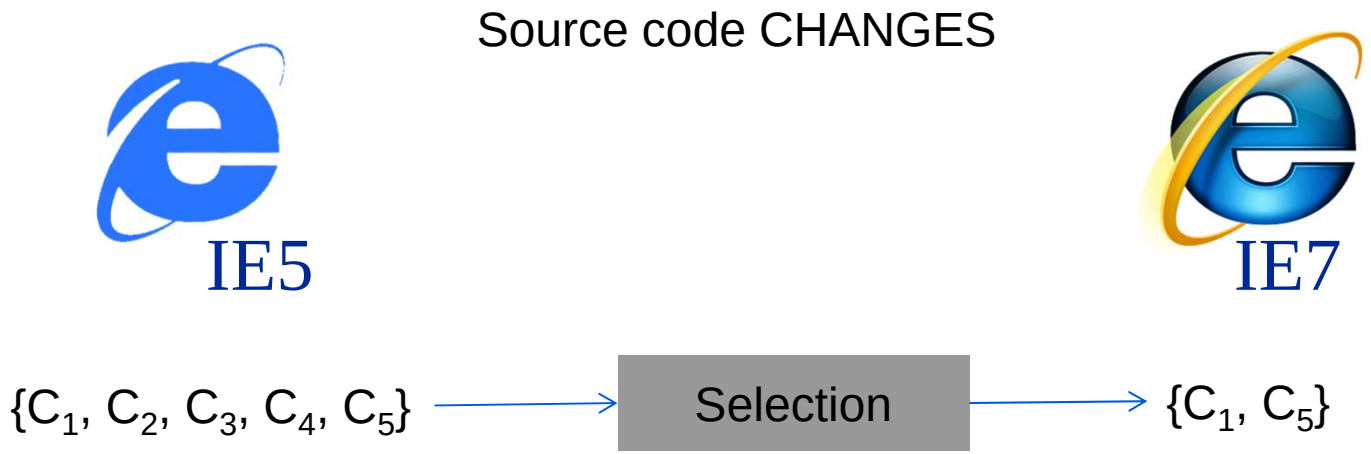
First configuration selection approach for regression testing

Source code CHANGES

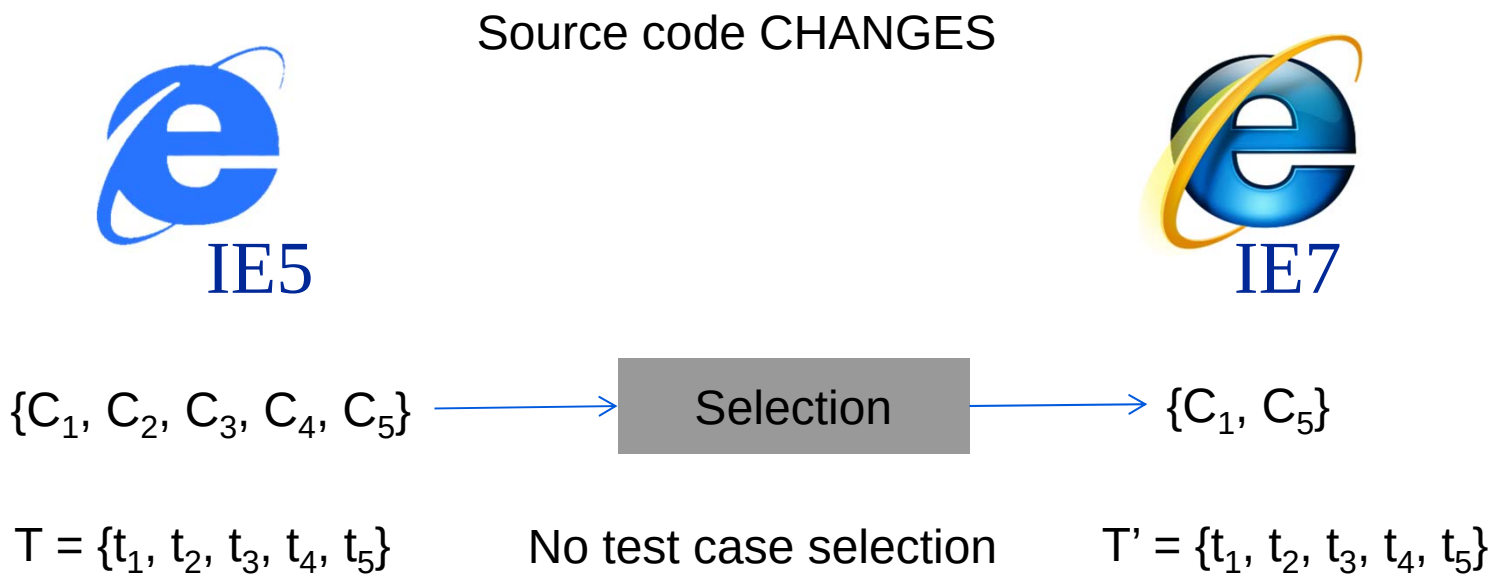
IE5

IE7

First configuration selection approach for regression testing



First configuration selection approach for regression testing



First configuration selection approach for regression testing

Source code CHANGES

$\{C_1, C_2, C_3, C_4, C_5\}$ → Selection → $\{C_1, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$ No test case selection $T' = \{t_1, t_2, t_3, t_4, t_5\}$

$\{C_1, C_5\}$ is both safe (wrt retest-all configurations) and non redundant

First configuration selection approach for regression testing

Source code CHANGES

IE5

IE7

$\{C_1, C_2, C_3, C_4, C_5\}$ → Selection → $\{C_1, C_5\}$

$T = \{t_1, t_2, t_3, t_4, t_5\}$ No test case selection $T' = \{t_1, t_2, t_3, t_4, t_5\}$

$\{C_1, C_5\}$ is both safe (wrt retest-all configurations) and non redundant

In our experiments, 15-60% of configurations were discarded as redundant saving 20-55% of regression testing time

Power and productivity
for a better world™