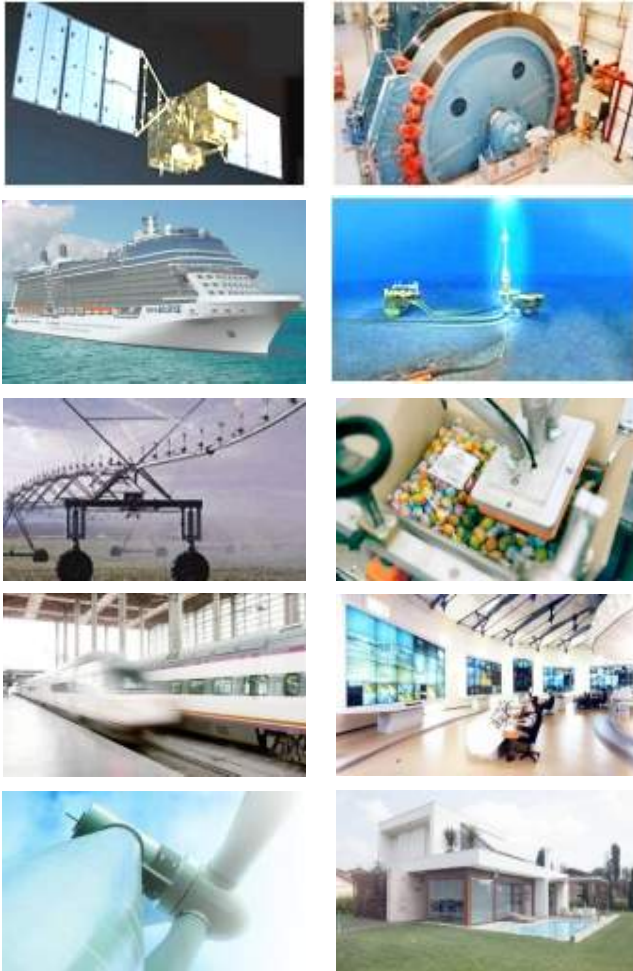




Code Change Impact Analysis for Testing Configurable Software Systems

Mithun Acharya
ABB Corporate Research
Raleigh NC USA

ABB: A power and automation company




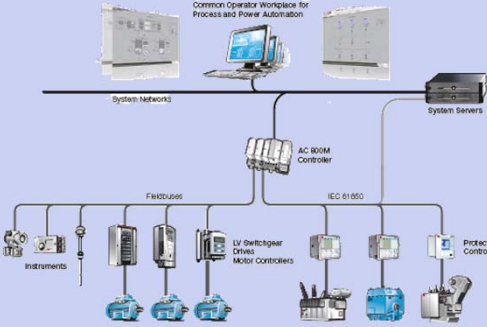
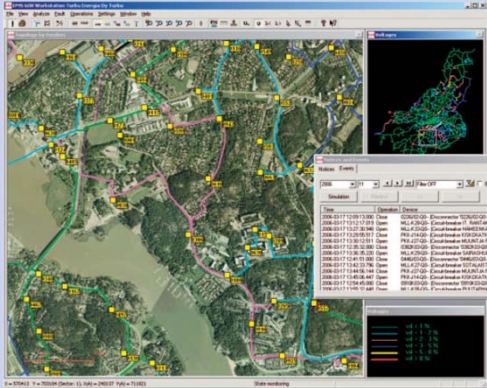
>125 years, >100 nations, ~150,000 employees

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

ABB Corporate Research Industrial Software Systems (ISS) research group



Software in ABB

Hardware with software inside	Software with few hardware components	Pure Software																																																												
		 <table border="1"><thead><tr><th>Time</th><th>Event</th><th>Source</th></tr></thead><tbody><tr><td>2008-01-17 12:08:13.000</td><td>Close</td><td>CB0401-01 - Environment 102401024</td></tr><tr><td>2008-01-17 12:12:23.000</td><td>Open</td><td>MS-21020 - Environment 102401024</td></tr><tr><td>2008-01-17 12:15:26.000</td><td>Close</td><td>MS-21020 - Environment 102401024</td></tr><tr><td>2008-01-17 12:16:05.071</td><td>Close</td><td>PS-41420 - Environment 102401024</td></tr><tr><td>2008-01-17 12:16:05.071</td><td>Close</td><td>PS-41420 - Environment 102401024</td></tr><tr><td>2008-01-17 12:20:20.000</td><td>Close</td><td>MSD01020 - Environment 102401024</td></tr><tr><td>2008-01-17 12:20:20.000</td><td>Close</td><td>MS-21020 - Environment 102401024</td></tr><tr><td>2008-01-17 12:21:10.000</td><td>Open</td><td>MSD01020 - Environment 102401024</td></tr><tr><td>2008-01-17 12:21:10.000</td><td>Open</td><td>MS-21020 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>MS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr><tr><td>2008-01-17 12:42:33.794</td><td>Close</td><td>PS-41210 - Environment 102401024</td></tr></tbody></table>	Time	Event	Source	2008-01-17 12:08:13.000	Close	CB0401-01 - Environment 102401024	2008-01-17 12:12:23.000	Open	MS-21020 - Environment 102401024	2008-01-17 12:15:26.000	Close	MS-21020 - Environment 102401024	2008-01-17 12:16:05.071	Close	PS-41420 - Environment 102401024	2008-01-17 12:16:05.071	Close	PS-41420 - Environment 102401024	2008-01-17 12:20:20.000	Close	MSD01020 - Environment 102401024	2008-01-17 12:20:20.000	Close	MS-21020 - Environment 102401024	2008-01-17 12:21:10.000	Open	MSD01020 - Environment 102401024	2008-01-17 12:21:10.000	Open	MS-21020 - Environment 102401024	2008-01-17 12:42:33.794	Close	MS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024	2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024
Time	Event	Source																																																												
2008-01-17 12:08:13.000	Close	CB0401-01 - Environment 102401024																																																												
2008-01-17 12:12:23.000	Open	MS-21020 - Environment 102401024																																																												
2008-01-17 12:15:26.000	Close	MS-21020 - Environment 102401024																																																												
2008-01-17 12:16:05.071	Close	PS-41420 - Environment 102401024																																																												
2008-01-17 12:16:05.071	Close	PS-41420 - Environment 102401024																																																												
2008-01-17 12:20:20.000	Close	MSD01020 - Environment 102401024																																																												
2008-01-17 12:20:20.000	Close	MS-21020 - Environment 102401024																																																												
2008-01-17 12:21:10.000	Open	MSD01020 - Environment 102401024																																																												
2008-01-17 12:21:10.000	Open	MS-21020 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	MS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												
2008-01-17 12:42:33.794	Close	PS-41210 - Environment 102401024																																																												

Software Evolution: A CSS constantly changes

pgmwpt_prev.c	pgmwpt_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 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, 30 FALSE, &vpt->tppcbot); 31 break; 32 } 33 } 34 (void) limitppc(task, vpt); 35 } 36 else 37 vpt->tppc = setppcwin(task, vpt->window, TRUE, L &vpt->tppcbot);	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, L &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, L FALSE, &vpt->tppcbot); 31 break; 32 } 33 } 34 (void) limitppc(task, vpt); 35 } 36 else 37 vpt->tppc = setppcwin(task, vpt->window, TRUE, L &vpt->tppcbot);

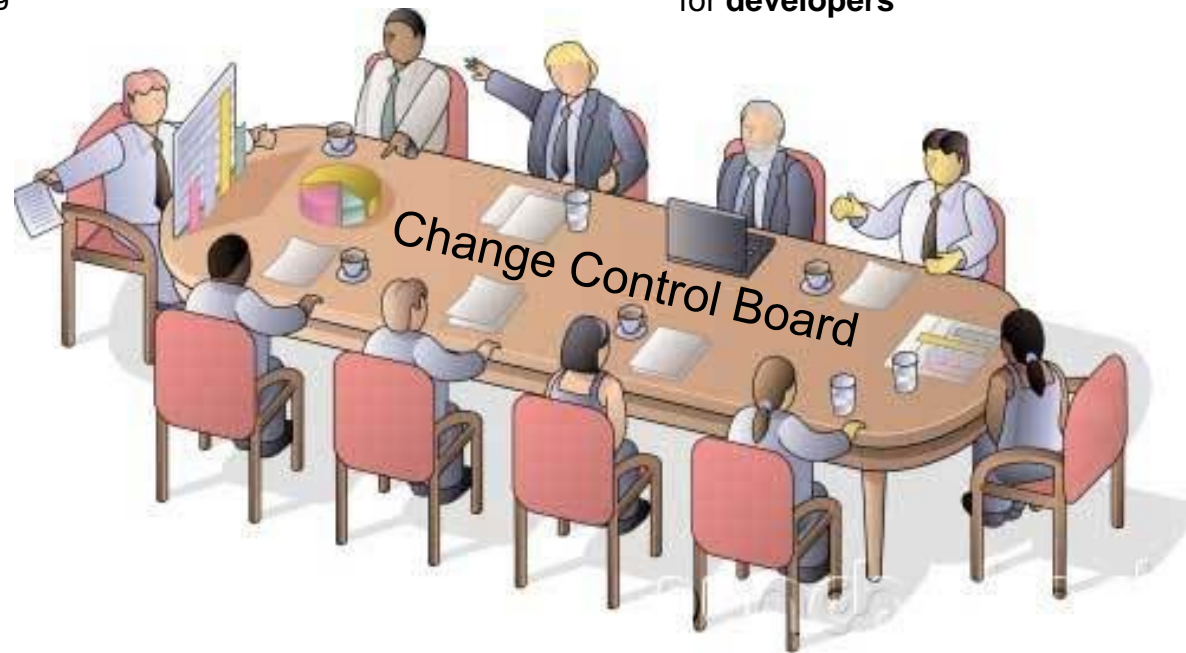
CHANGE

Hundreds of such changes committed daily

Change Control Board meetings

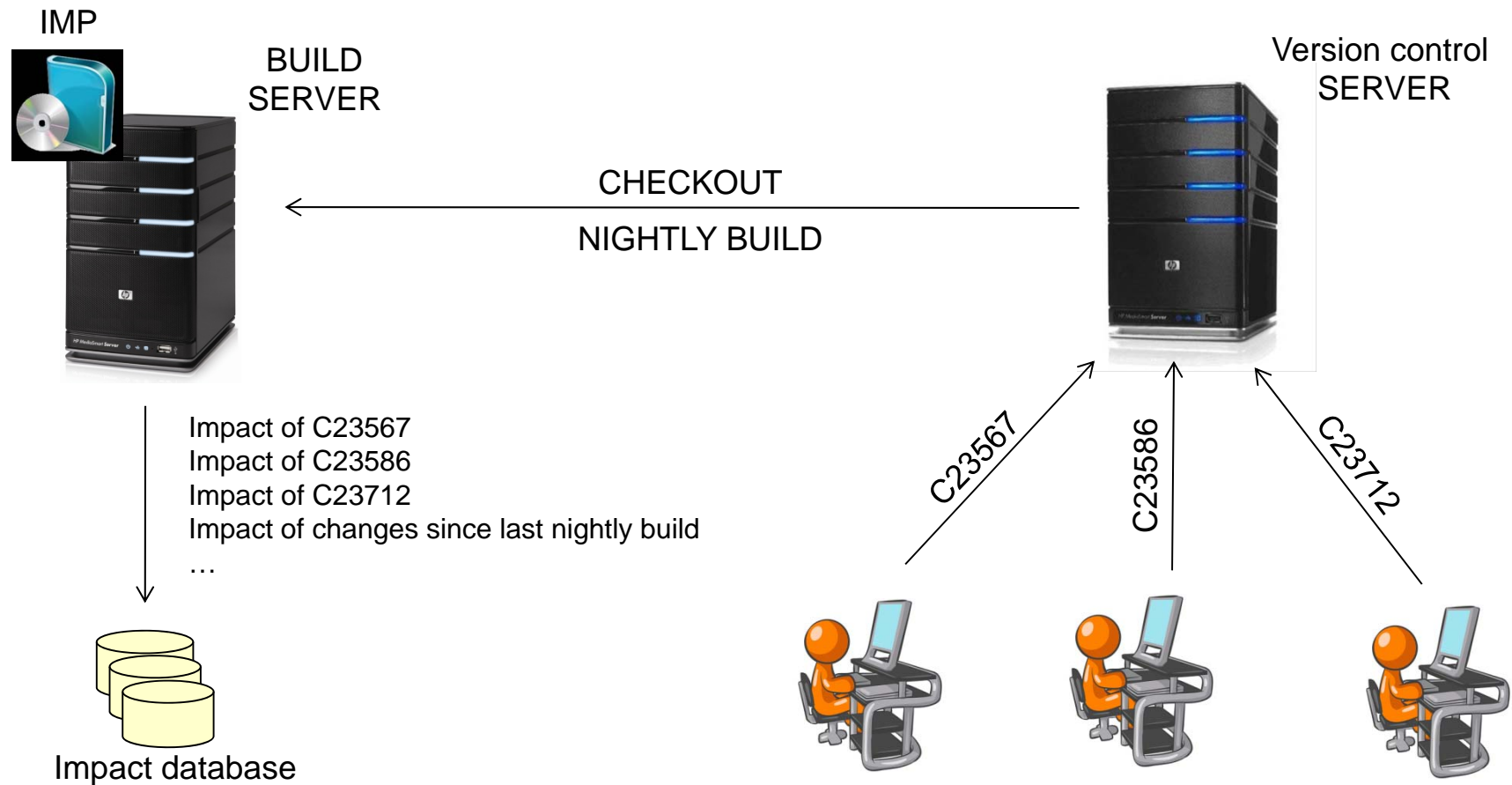
Change impact visualizations
for **managers** for decision
making

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



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

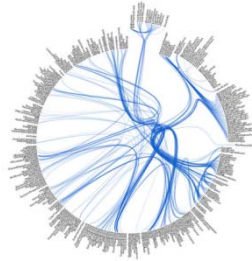
Imp: Code change impact analysis for C/C++ programs



Quantifiable risk/cost analysis of changes to CSS

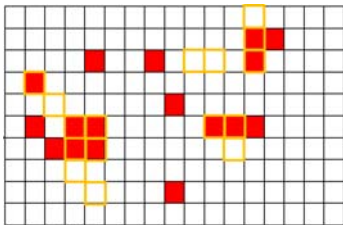
Automated Dependency Analysis

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?

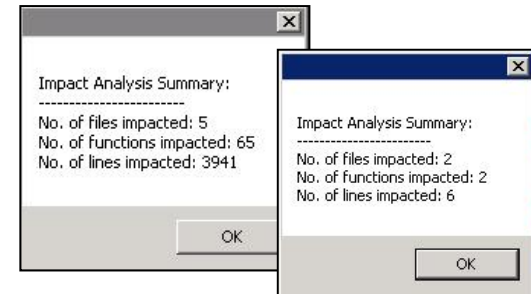


Overlay change impact with risky areas in code



Automated What-If Analysis

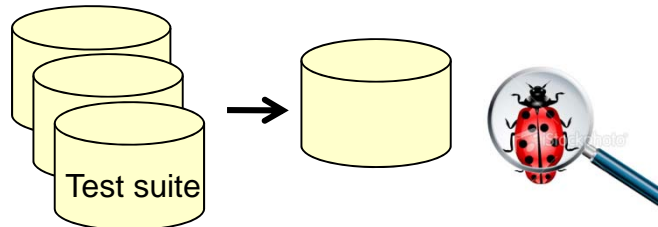
What is the 'best' way to fix this bug or implement that new feature?



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);
}

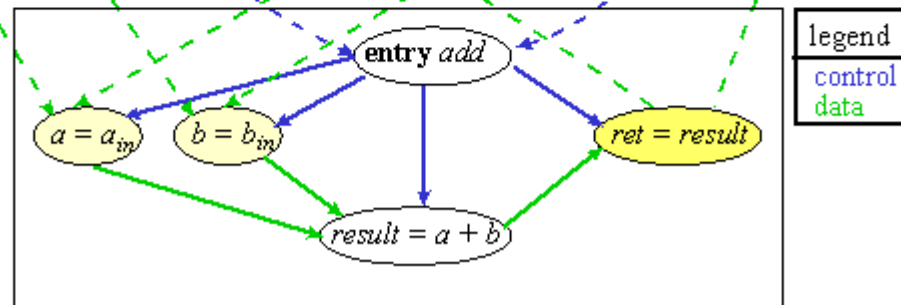
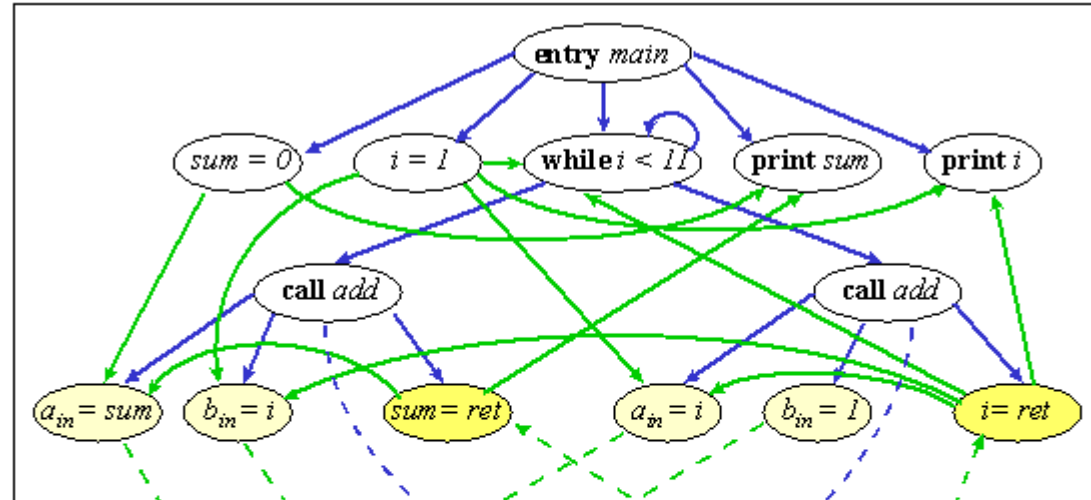
```

```

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

```

Program Dependence Graph (PDG) for main



PDG for add

legend
 control
 data

System Dependence Graph (SDG)

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 Acharya, 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

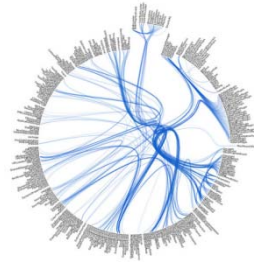
Tingting Yu, Xiao Qu, Mithun Acharya, Gregg Rothermel. *Oracle-Based Regression Test Selection*. *Under submission*.

Regression test selection

Regression testing of CSS with code change impact analysis

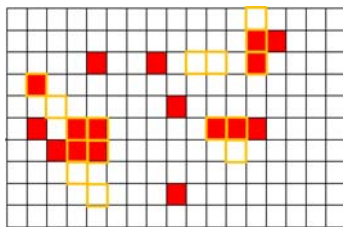
Automated Dependency Analysis

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



Automated Risk/Cost Analysis

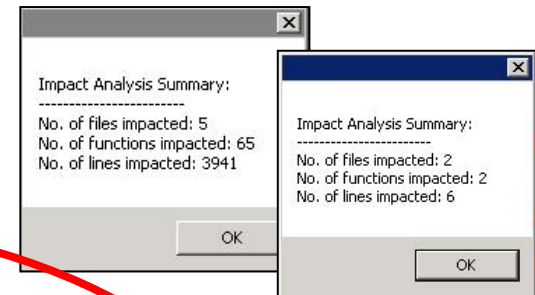
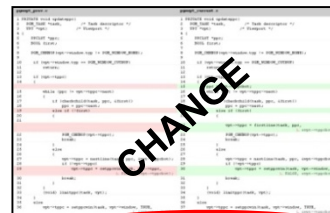
3 days to release!!! Should I implement this feature or bug fix?



Overlay change impact with risky areas in code

Automated What-If Analysis

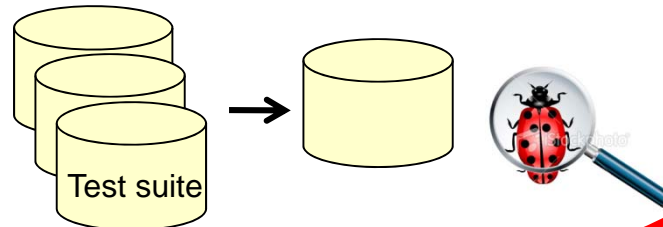
What is the 'best' way to fix this bug or implement that new feature?



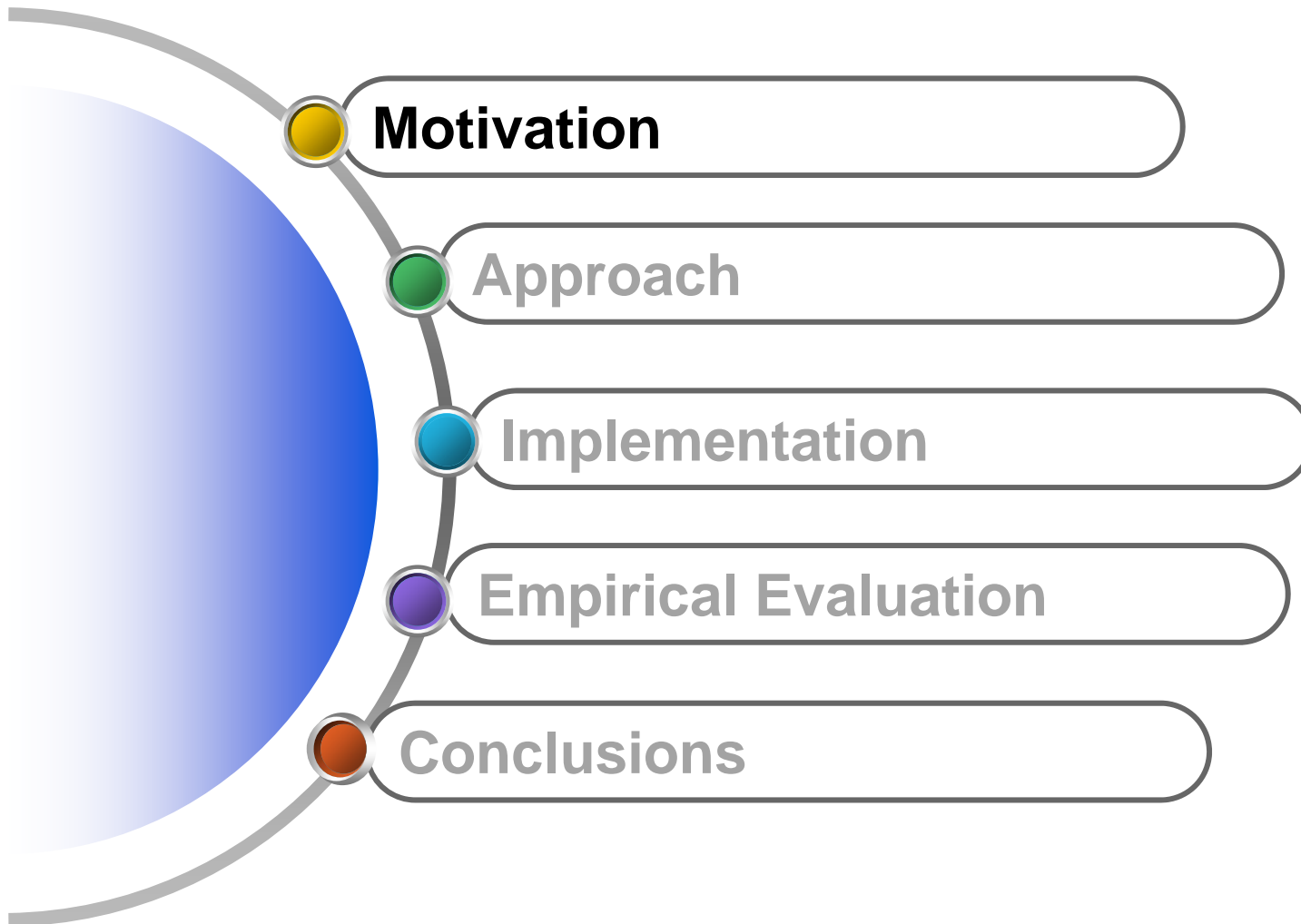
3941 lines vs. 6 lines

Automated Regression Testing

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

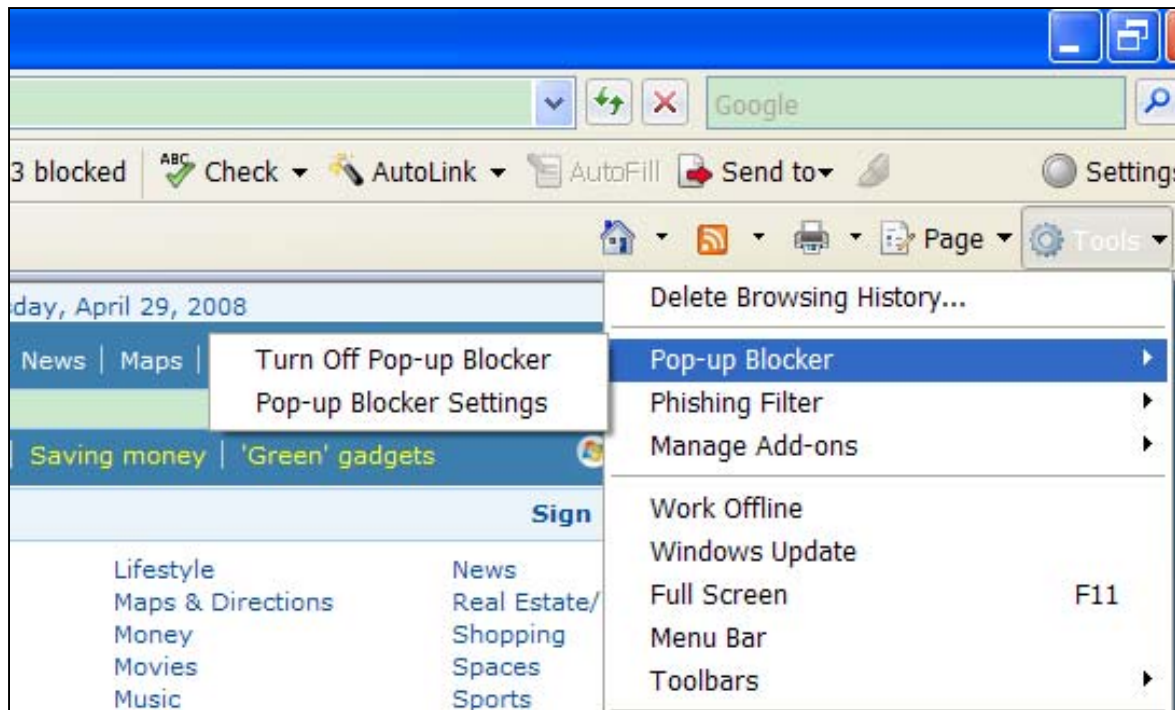


Outline



Configurable Software Systems

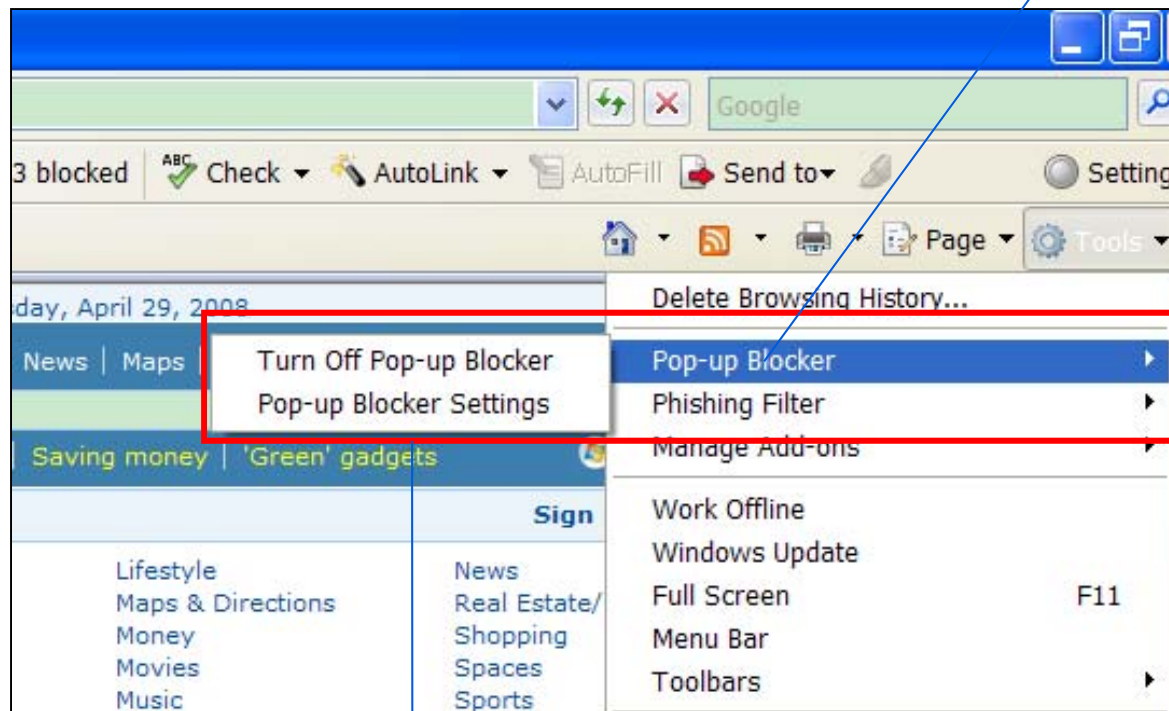
- Software that can be customized through a set of options
- Example: [Internet Explorer](#)



Configurable Software Systems

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

Configurable option: **“Pop-up Blocker”**




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	...
...

Option



Internet Explorer Configurations

The diagram illustrates a table of Internet Explorer configurations. The table has four columns and five rows. The first column lists the configuration options, and the subsequent columns show the values for two different configurations, labeled C_1 and C_2 , followed by an ellipsis. Blue arrows point from the labels 'Option' and 'Value' to the corresponding parts of the table.

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

Internet Explorer Configurations

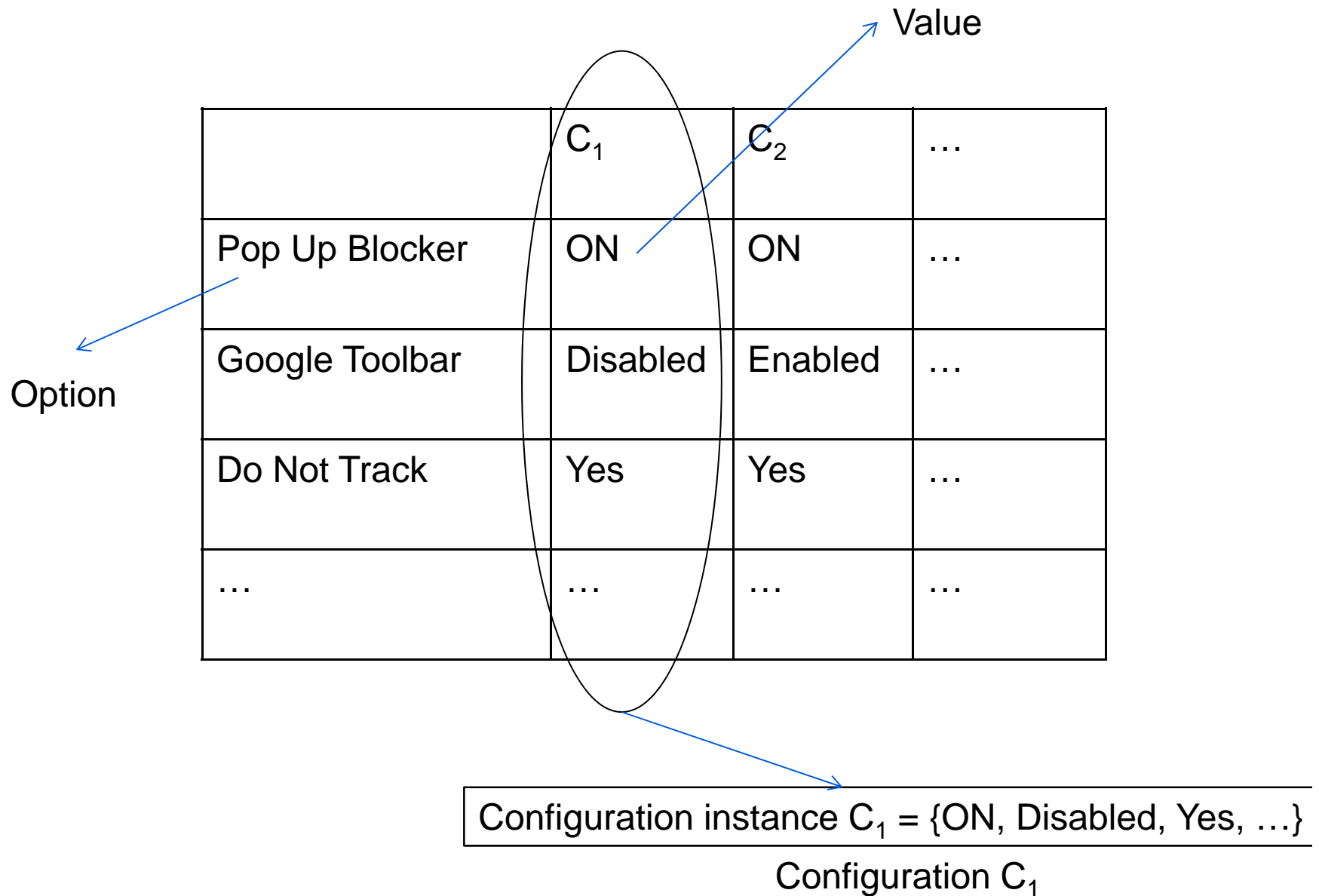
	C_1	C_2	...
Pop Up Blocker	ON	ON	...
Google Toolbar	Disabled	Enabled	...
Do Not Track	Yes	Yes	...
...

Option

Value

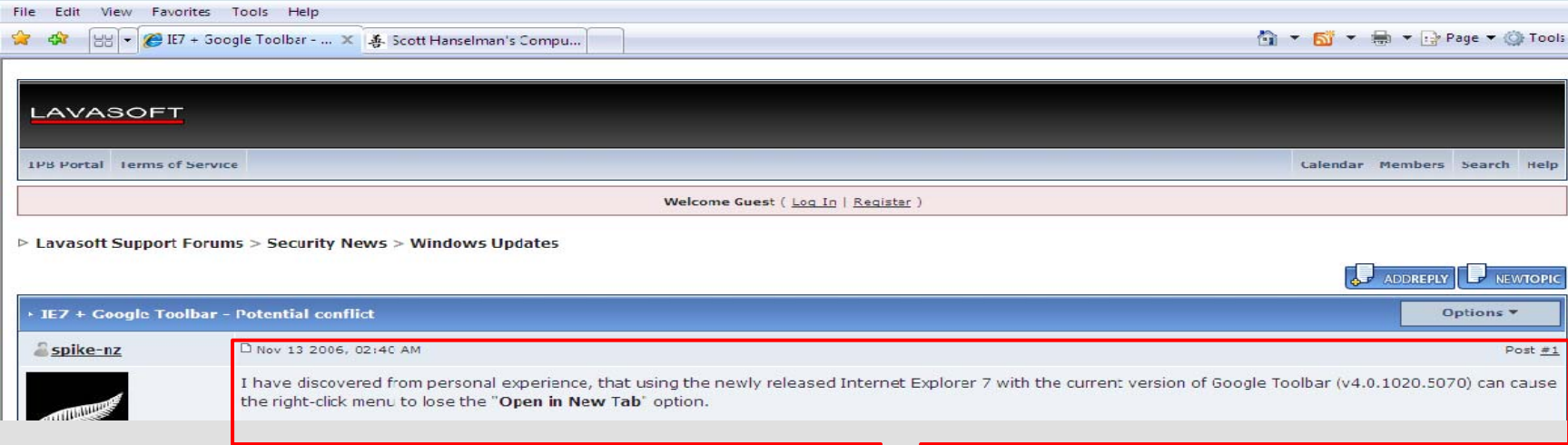
Configuration instance $C_1 = \{ON, Disabled, Yes, \dots\}$

Internet Explorer Configurations



Impact of Configurations on System Behavior

Faulty System Behavior under Certain Configurations



The screenshot shows an Internet Explorer 7 browser window with the address bar displaying "IE7 + Google Toolbar - ...". The page content includes the Lavasoft logo, navigation links like "IPB Portal" and "Terms of Service", and a forum thread titled "IE7 + Google Toolbar - Potential conflict". A red box highlights a post by user "spike-nz" dated "Nov 13 2006, 02:14C AM". The post text reads: "I have discovered from personal experience, that using the newly released Internet Explorer 7 with the current version of Google Toolbar (v4.0.1020.5070) can cause the right-click menu to lose the 'Open in New Tab' option." A red arrow points from this post to the text below.

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

File Edit View Favorites Tools Help

IE7 + Google Toolbar - ... X Scott Hanselman's Compu...

LAVASOFT

IPB Portal Terms of Service Calendar Members Search Help

Welcome Guest ([Log In](#) | [Register](#))

► Lavasoft Support Forums > Security News > Windows Updates

ADDREPLY NEWTOPIC Options

IE7 + Google Toolbar - Potential conflict

spike-nz Nov 13 2006, 02:14C AM Post #1

I have discovered from personal experience, that using the newly released Internet Explorer 7 with the current version of Google Toolbar (v4.0.1020.5070) can cause the right-click menu to lose the "Open in New Tab" option.

Until now, to open a new Tab in the same window, I had to hold down "Ctrl" whilst clicking on the Link.

For those of you who have encountered this problem, open IE7, click "Tools > Manage Add-ons > Enable or Disable Add-ons"

Gr
Ad
Po
Jo
Frc
Ze
Me

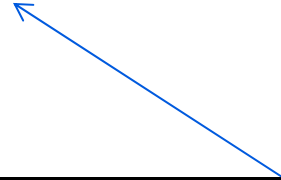
To fix this, open IE7, click "Tools > Manage Add-ons > **Disable Google Toolbar**"

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



```
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
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

```



```
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
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



```
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
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

```



```
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
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

```

$C_1 = \{\text{ON, disabled, Yes, ...}\}$

Configurations control system execution



Test: PASS



```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token
  no_login_required
  cache_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

```

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token
  no_login_required
  cache_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

```

$C_1 = \{ON, disabled, Yes, \dots\}$

Configurations control system execution



Test: PASS



```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token
  no_login_required
  cache_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

```

$C_1 = \{ON, disabled, Yes, \dots\}$

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token
  no_login_required
  cache_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

```

$C_2 = \{ON, enabled, Yes, \dots\}$

Configurations control system execution



Test: PASS

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token
  no_login_required
  cache_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

```

$C_1 = \{ON, disabled, Yes, \dots\}$



Test: FAIL

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token
  no_login_required
  cache_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

```

$C_2 = \{ON, enabled, Yes, \dots\}$

Configurations control system execution



Test: PASS

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token
  no_login_required
  cache_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

```

$C_1 = \{ON, disabled, Yes, \dots\}$



Test: FAIL

```
class SiteController < ApplicationController
  skip_before_filter :verify_authenticity_token
  no_login_required
  cache_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

```

$C_2 = \{ON, enabled, Yes, \dots\}$

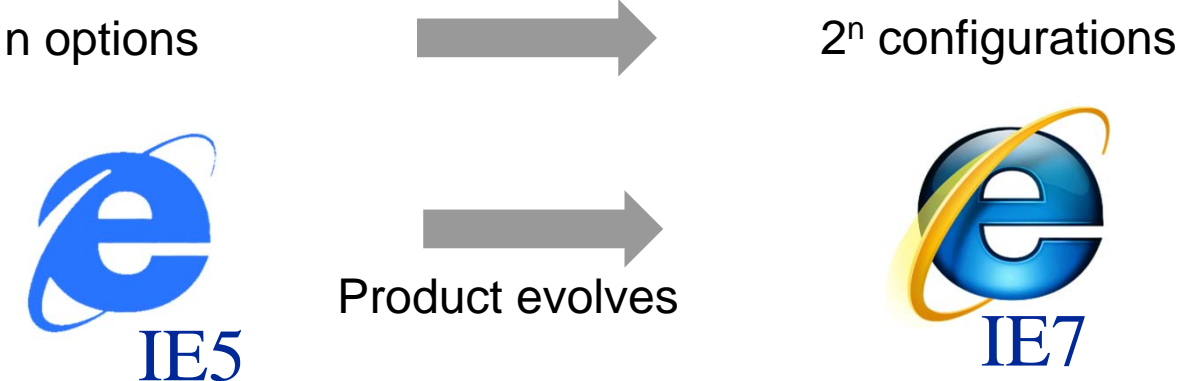
Can we statically approximate how configurations (options) control system execution?

Challenges for testing configurable systems

Challenges for testing configurable systems



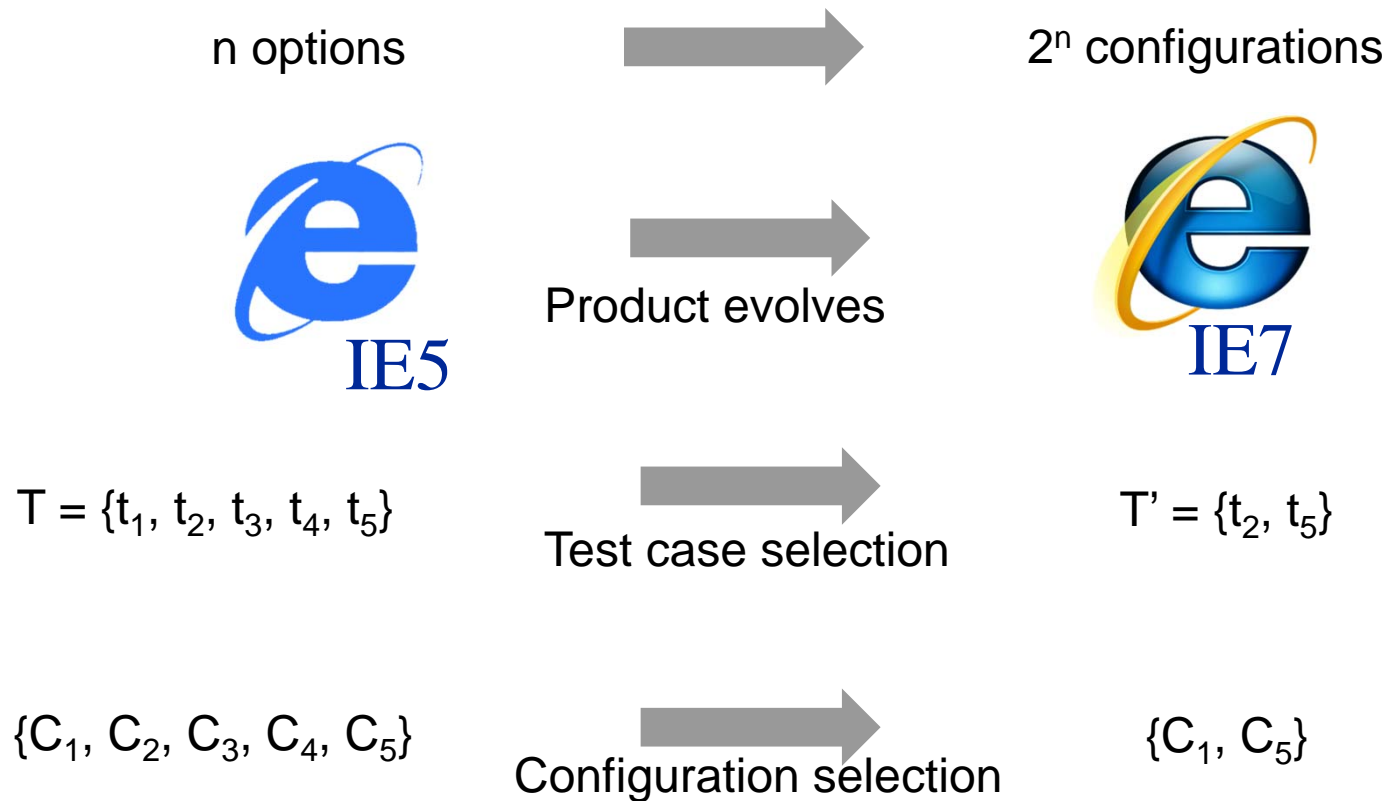
Challenges for testing configurable systems



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



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



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



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



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



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

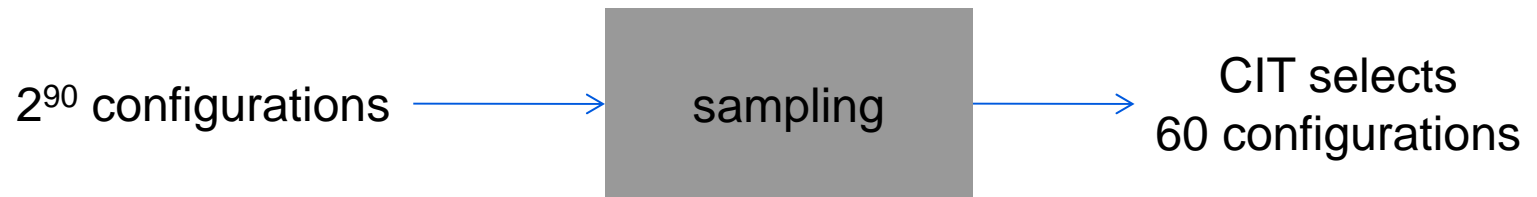
We choose to test IE7 only under sampled configurations $C_1, C_3,$ and C_4 and for each configurations we test IE7 with all tests $\{t_1, t_2, t_3, t_4, t_5\}$

`vim`: A configurable system

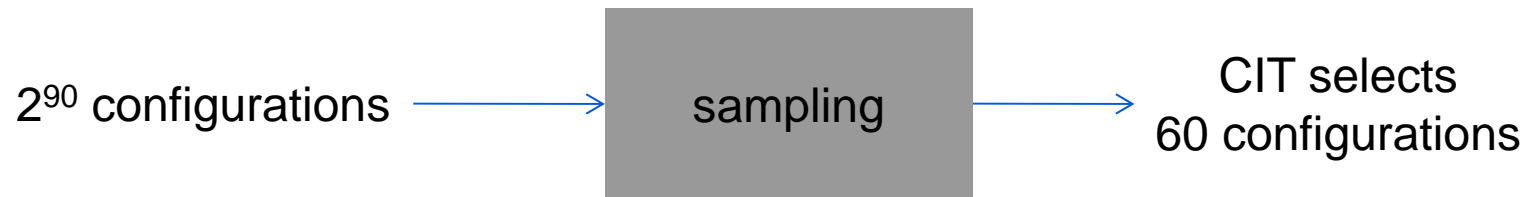
vim: A configurable system

2^{90} configurations

vim: A configurable system

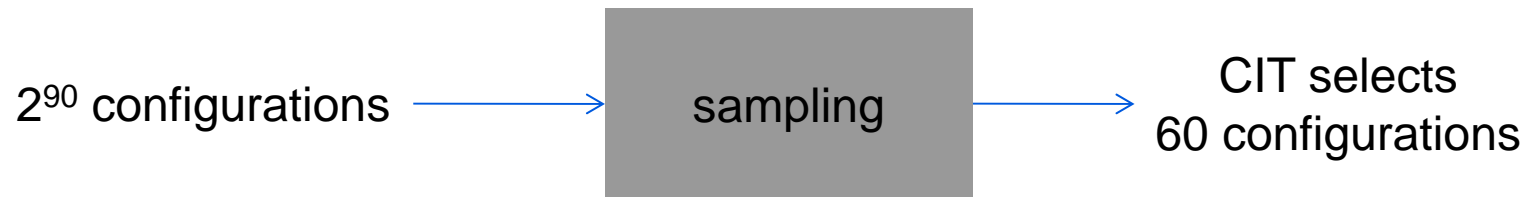


vim: A configurable system



Rerun the full test suite on each 60 configurations

vim: A configurable system

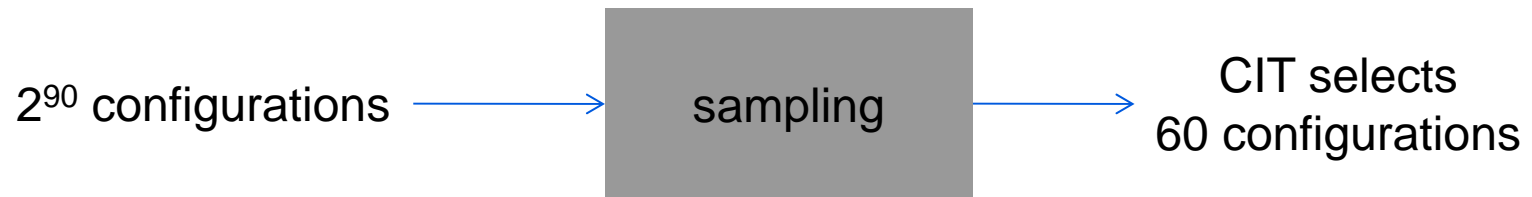


Rerun the full test suite on each 60 configurations

7 hours to execute the full test suite

Takes $7 \cdot 60 = 420$ 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 $7 \cdot 60 = 420$ 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*

Test case selection when configuration under test changes*



Test case selection when configuration under test changes*

Source code DOES NOT change



Configuration under test changes

$C_1 = \{\text{ON, Disabled, Yes, ...}\}$

$C_2 = \{\text{ON, Enabled, Yes, ...}\}$

Test case selection when configuration under test changes*

Source code DOES NOT change



Configuration under test changes

$C_1 = \{\text{ON, Disabled, Yes, ...}\}$

$C_2 = \{\text{ON, Enabled, Yes, ...}\}$

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

$C_1 = \{\text{ON, Disabled, Yes, ...}\}$

$C_2 = \{\text{ON, Enabled, Yes, ...}\}$

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

$C_1 = \{\text{ON, Disabled, Yes, ...}\}$

$C_2 = \{\text{ON, Enabled, Yes, ...}\}$

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

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

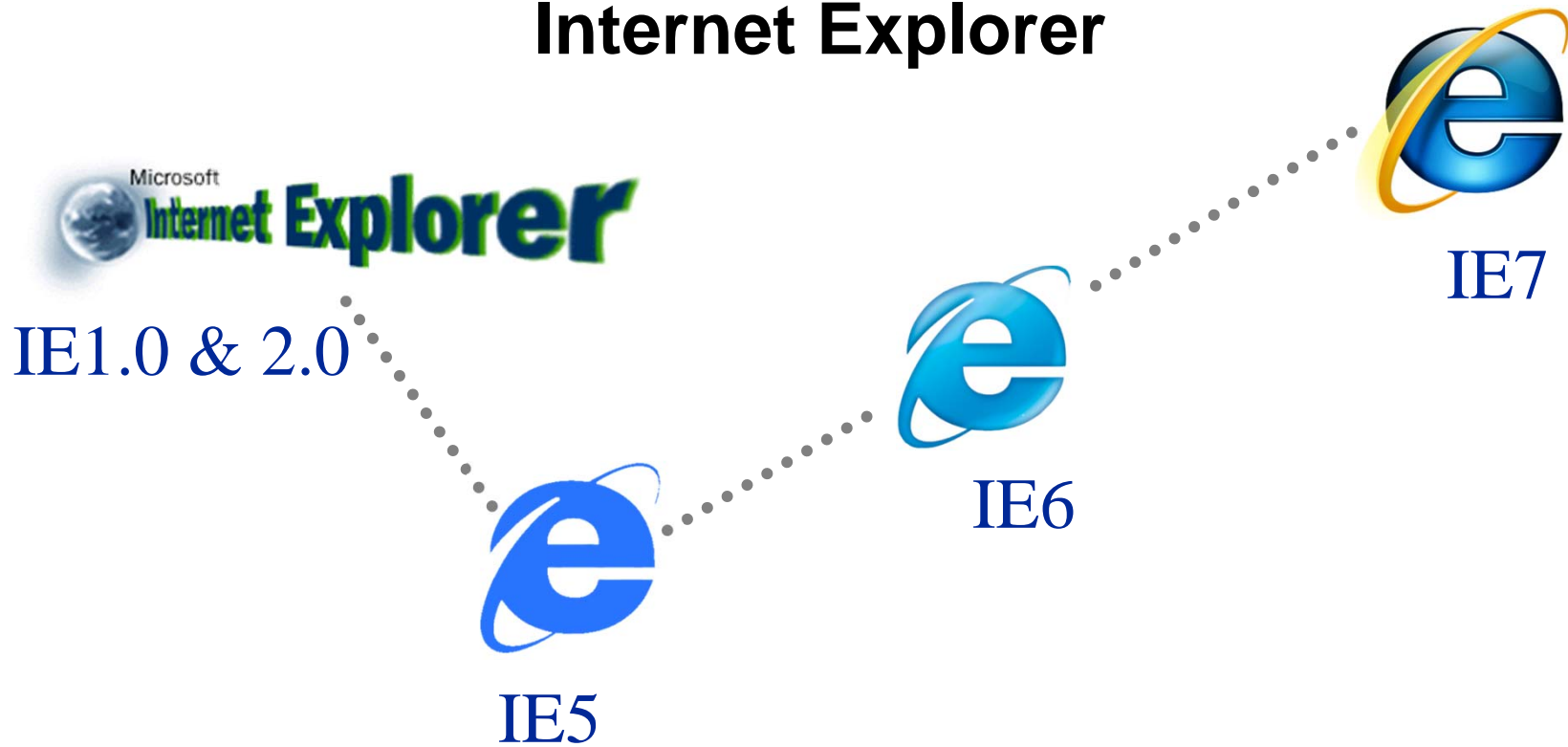
$T' = \{t_2, t_5\}$

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

Internet Explorer



Configuration prioritization for regression testing*

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



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

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

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

Configuration prioritization for regression testing*

Source code CHANGES

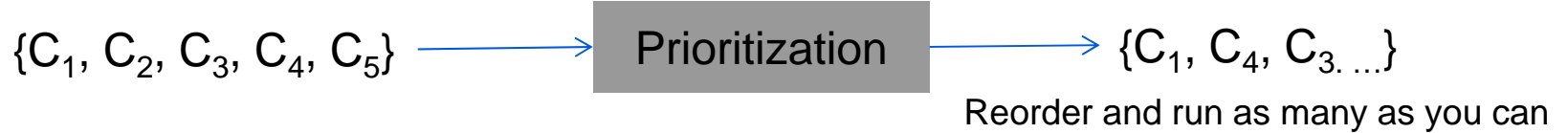


$$T = \{t_1, t_2, t_3, t_4, t_5\}$$

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

Configuration prioritization for regression testing*

Source code CHANGES

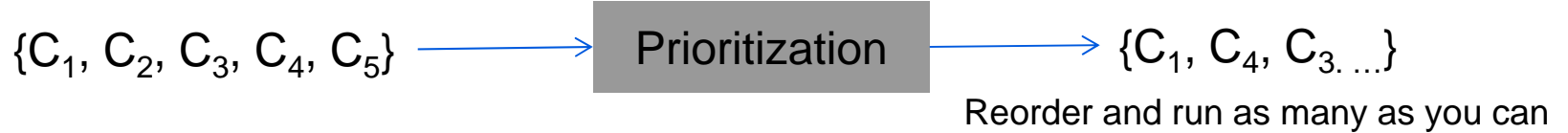


$$T = \{t_1, t_2, t_3, t_4, t_5\}$$

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

Configuration prioritization for regression testing*

Source code CHANGES



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

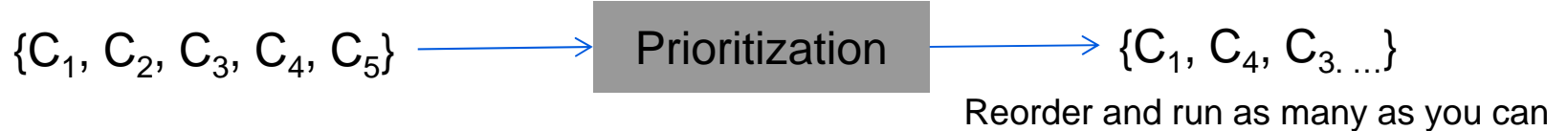
No test case selection

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

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

Configuration prioritization for regression testing*

Source code CHANGES



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

No test case selection

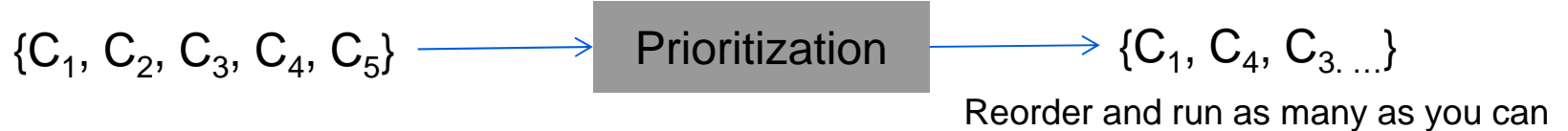
$T' = \{t_1, t_2, t_3, t_4, t_5\}$

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



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

No test case selection

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

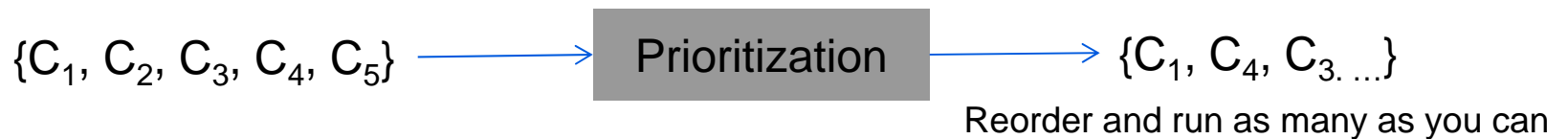
Increases rate of fault detection. But...

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



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

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

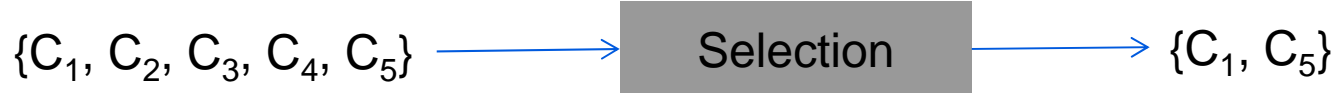


$\{C_1, C_2, C_3, C_4, C_5\}$

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

Configuration selection for regression testing (Focus of this talk)

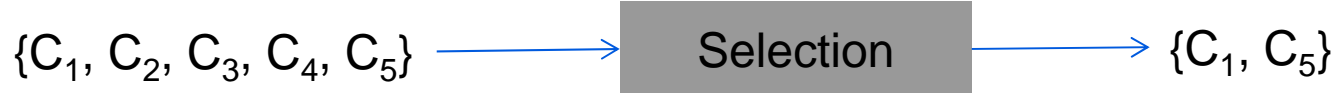
Source code CHANGES



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

Configuration selection for regression testing (Focus of this talk)

Source code CHANGES



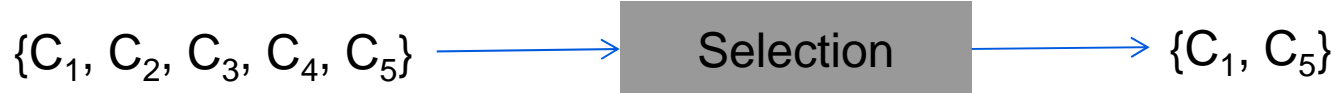
$T = \{t_1, t_2, t_3, t_4, t_5\}$

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



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

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

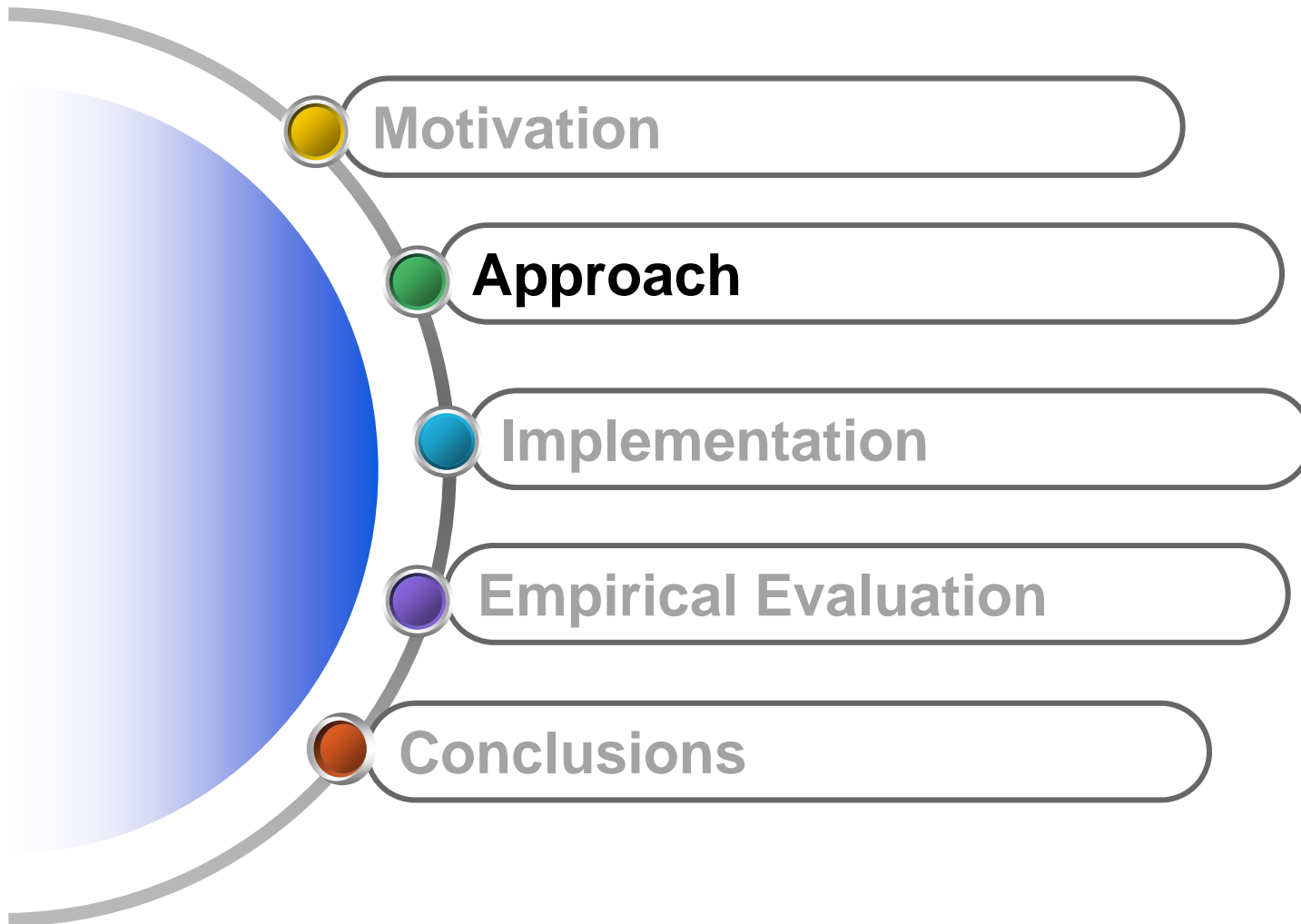
Problems		Single Version Testing	Regression Testing
Configuration Level	Selection	[11][15][18][25]	Focus of this talk
	Prioritization	NA	[18]
Test Case Level	Selection	[17]	[22][23] ^a
	Prioritization	NA	[21] ^a

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

Outline



Key Idea: Map configuration options to code

Key Idea: Map configuration options to code

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



```
254 void TestGetRenderingInfo() {
255 // Initialize the state for the test, the test and the target
256 InitializeTest();
257 active = false;
258 target_name = target;
259 target_url = url;
260 target_renderer = Renderer;
261 // Merge starting conditions and options for the target and the test
262 target_renderer.MergeOptions(target_renderer);
263 // Calculate the mapping between the target and the test
264 // Calculate the mapping between the target and the test
265 // Map the output
266 #if (WIN_ARCHITECTURE == 32) || (WIN_ARCHITECTURE == 64)
267     MapOutput();
268 #endif
269 // Test the test (1) Test the test
270 mapping = 0;
271 do {
272 // Update the current step
273     Update();
274 // Check the current status of the target and subsequent status of
275 // (target_url) target_url=target_url;
276 //
277 for (counter = 0; counter < MAX_COUNTER; counter++) {
278 // (counter) counter;
279 //
280 // Calculate the test results
281     CalculateTestResults();
282 // Calculate the test results
283     CalculateTestResults();
284 // Map the output
285 #if (WIN_ARCHITECTURE == 32) || (WIN_ARCHITECTURE == 64)
286     MapOutput();
287 #endif
288 }
```

Key Idea: Map configuration options to code

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



```
215 void Team::performSingleTest() {
216 // Initialize the state for the MAVs, the team and the target
217 initializeMavs();
218 alive = true;
219 target_alive = false;
220 target_damaged = false;
221 target_stepsOfDamageIncurred = 0;
222
223 // Assign starting coordinates and angles to the target and to the MAVs
224 target.calculateStartingCoordinates();
225 calculateMavsStartingCoordinates();
226 calculateMavsStartingAngles();
227
228 // Calculate the starting distance between the target and the MAVs
229 calculateTargetMavDistance();
230
231 // Empty the buffers
232 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
233 emptyMavBuffers();
234 #endif
235
236 // Test the team till there are MAVs alive
237 currentStep = 0;
238 do {
239 // Decrease the current step
240 currentStep--;
241
242 // Check the current status of the target and subsequently restore it
243 if (target_damaged) target_restoreFromDamage(currentStep);
244
245 for (currentMav = 0; currentMav < MAVS_PER_TEAM; currentMav++) {
246 if (isMav[currentMav]->alive) {
247 // Calculate MAV's perception
248 calculateMavPerception();
249
250 // Calculate MAV's outputs
251 calculateMavOutputs();
252
253 // Fill the buffers
254 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
255 fillMavBuffers();
256 #endif
257 }
258 }
```



```
214 void Team::performSingleTest() {
215 // Initialize the state for the MAVs, the team and the target
216 initializeMavsState();
217 alive = true;
218 target_alive = false;
219 target_damaged = false;
220 target_stepsOfDamageIncurred = 0;
221
222 // Assign starting coordinates and angles to the target and to the MAVs
223 target.calculateStartingCoordinates();
224 calculateMavsStartingCoordinates();
225 calculateMavsStartingAngles();
226
227 // Calculate the starting distance between the target and the MAVs
228 calculateTargetMavDistance();
229
230 // Empty the buffers
231 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
232 emptyMavBuffers();
233 #endif
234
235 // Test the team till there are MAVs alive
236 currentStep = 0;
237 do {
238 // Decrease the current step
239 currentStep--;
240
241 // Check the current status of the target and subsequently restore it
242 if (target_damaged) target_restoreFromDamage(currentStep);
243
244 for (currentMav = 0; currentMav < MAVS_PER_TEAM; currentMav++) {
245 if (isMav[currentMav]->alive) {
246 // Calculate MAV's perception
247 calculateMavPerception();
248
249 // Calculate MAV's outputs
250 calculateMavOutputs();
251
252 // Fill the buffers
253 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
254 fillMavBuffers();
255 #endif
256 }
257 }
```

change

Key Idea: Map configuration options to code

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



```
215 void TestGetPerception() {
216 // Initialize the state for the MAVs, the team and the target
217 InitializeState();
218 alive = true;
219 target_alive = false;
220 target_damaged = false;
221 target->lossOfDamageIncurred = 0;
222 // Assign starting coordinates and angles to the target and to the MAVs
223 target->assignStartingCoordinates();
224 calculateMAVStartingCoordinates();
225 calculateMAVStartingAngles();
226 // Calculate the starting distance between the target and the MAVs
227 calculateTargetMAVDistance();
228 // Empty the buffers
229 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
230 emptyMAVBuffers();
231 #endif
232 // Test the team till there are MAVs alive
233 currentStep = 0;
234 while (
235 // Decrease the current step
236 currentStep > 0)
237 // Check the current status of the target and subsequently restore it
238 if (target->isDamaged() || target->isInPerception(currentStep))
239 for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
240 if (stateMAV[currentMAV]->alive) {
241 // Calculate MAV's perception
242 calculateMAVPerception();
243 // Calculate MAV's outputs
244 MAVOutputs();
245 // Fill the buffers
246 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
247 fillMAVBuffers();
248 #endif
249 }
250 }
```



```
214 void TestGetPerception() {
215 // Initialize the state for the MAVs, the team and the target
216 InitializeState();
217 alive = true;
218 target_alive = false;
219 target_damaged = false;
220 target->lossOfDamageIncurred = 0;
221 // Assign starting coordinates and angles to the target and to the MAVs
222 target->assignStartingCoordinates();
223 calculateMAVStartingCoordinates();
224 calculateMAVStartingAngles();
225 // Calculate the starting distance between the target and the MAVs
226 calculateTargetMAVDistance();
227 // Empty the buffers
228 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
229 emptyMAVBuffers();
230 #endif
231 // Test the team till there are MAVs alive
232 currentStep = 0;
233 while (
234 // Decrease the current step
235 currentStep > 0)
236 for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
237 if (stateMAV[currentMAV]->alive) {
238 // Calculate MAV's perception
239 calculateMAVPerception();
240 // Calculate MAV's outputs
241 MAVOutputs();
242 // Fill the buffers
243 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
244 fillMAVBuffers();
245 #endif
246 }
247 }
```

change

Key Idea: Map configuration options to code

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



```
214 void TestPerformingTest() {
215 // Initialize the state for the MAV, the team and the target
216 InitializeState();
217 alive = true;
218 target_alive = false;
219 target_damaged = false;
220 target_damage_incurred = 0;
221 // Align starting coordinates and angles to the target and to the MAV
222 target->alignStartingCoordinates();
223 calculateMAVStartingAngles();
224 // Calculate the starting distance between the target and the MAV
225 calculateTargetMAVDistance();
226 // Empty the buffers
227 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
228 emptyMAVbuffers();
229 #endif
230 // Test the team till there are MAVs alive
231 currentStep = 0;
232 while (true) {
233 // Decrease the current step
234 currentStep--;
235 // Check the current status of the target and subsequent actions if
236 if (target_damaged || target_alive || currentStep) {
237 for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
238 if (state[currentMAV] == alive) {
239 // Calculate MAV's perception
240 calculateMAVPerception();
241 // Calculate MAV's outputs
242 calculateMAVOutputs();
243 // Fill the buffers
244 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
245 fillMAVbuffers();
246 #endif
247 }
248 }
249 }
250 }
```



```
214 void TestPerformingTest() {
215 // Initialize the state for the MAV, the team and the target
216 InitializeState();
217 alive = true;
218 target_alive = false;
219 target_damaged = false;
220 target_damage_incurred = 0;
221 // Align starting coordinates and angles to the target and to the MAV
222 target->alignStartingCoordinates();
223 calculateMAVStartingAngles();
224 // Calculate the starting distance between the target and the MAV
225 calculateTargetMAVDistance();
226 // Empty the buffers
227 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
228 emptyMAVbuffers();
229 #endif
230 // Test the team till there are MAVs alive
231 currentStep = 0;
232 while (true) {
233 // Decrease the current step
234 currentStep--;
235 // Check the current status of the target and subsequent actions if
236 if (target_damaged || target_alive || currentStep) {
237 for (currentMAV = 0; currentMAV < MAVS_PER_TEAM; currentMAV++) {
238 if (state[currentMAV] == alive) {
239 // Calculate MAV's perception
240 calculateMAVPerception();
241 // Calculate MAV's outputs
242 calculateMAVOutputs();
243 // Fill the buffers
244 #if (NL_ARCHITECTURE == 13) || (NL_ARCHITECTURE == 24)
245 fillMAVbuffers();
246 #endif
247 }
248 }
249 }
250 }
```

change

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         emptyMAVBuffers();
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 configuration option impact

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

```
214 // Perform a single test 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->stepIfDamageIncurred = 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         emptyMAVBuffers();
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
261             }
262         }
263     } while (currentStep <= MAX_STEPS);
264 }
```



Key Idea: statically compute configuration option impact

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

```
214 void Team::performSingleTest() {
215
216
217 // Initialize the stats for the MAVs, the team and the target
218 initializeMAVStats();
219 alive = true;
220 target->alive = true;
221 target->damaged = false;
222 target->stepIfDamageIncurred = 0;
223
224 // Assign starting coordinates and angles to the target and to the MAVs
225 target->calculateStartingCoordinates();
226 calculateMAVStartingCoordinates();
227 calculateMAVStartingAngles();
228
229 // Calculate the starting distance between the target and the MAVs
230 calculateTargetMAVDistance();
231
232 // Empty the buffers
233 #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234     emptyMAVBuffers();
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 configuration option impact

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

```
214 void Team::performSingleTest() {
215
216
217 // Initialize the stats for the MAVs, the team and the target
218 initializeMAVStats();
219 alive = true;
220 target->alive = true;
221 target->damaged = false;
222 target->stepIfDamageIncurred = 0;
223
224 // Assign starting coordinates and angles to the target and to the MAVs
225 target->calculateStartingCoordinates();
226 calculateMAVStartingCoordinates();
227 calculateMAVStartingAngles();
228
229 // Calculate the starting distance between the target and the MAVs
230 calculateTargetMAVDistance();
231
232 // Empty the buffers
233 #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234     emptyMAVBuffers();
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

```

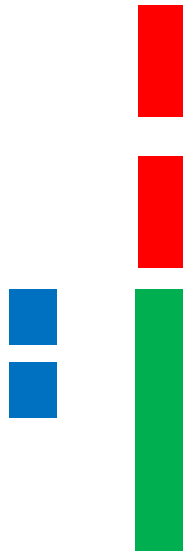


Key Idea: statically compute impact of the changes

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

```
214 void Team::performSingleTest() {
215
216
217 // Initialize the stats for the MAVs, the team and the target
218 initializeMAVStats();
219 alive = true;
220 target->alive = true;
221 target->damaged = false;
222 target->stepIfDamageIncurred = 0;
223
224 // Assign starting coordinates and angles to the target and to the MAVs
225 target->calculateStartingCoordinates();
226 calculateMAVStartingCoordinates();
227 calculateMAVStartingAngles();
228
229 // Calculate the starting distance between the target and the MAVs
230 calculateTargetMAVDistance();
231
232 // Empty the buffers
233 #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234     emptyMAVBuffers();
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         if (simMAV[currentMAV]->alive) {
249
250             // Calculate MAV's perception
251             calculateMAVPerception();
252
253             // Calculate NN's outputs
254             n_net->NN->step();
255
256             // Fill the buffers
257             #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
258                 fillMAVBuffers();
259             #endif
260

```



Key Idea: Intersect configuration impact with change impact

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

The diagram illustrates the intersection of configuration and change impacts. It features a central code block with a black background and white text. To the left, there are two vertical bars: a red one and a green one. To the right, there is a blue 'e' logo with 'IE7' below it, and an orange vertical bar. A blue arrow points from the configuration options box to the code block. A red arrow points from the configuration options box to the code block. A green arrow points from the configuration options box to the code block. A blue arrow points from the code block to the IE7 logo. A black oval encircles the code block and the orange bar. The code block contains the following text:

```
214 void Team::performSingleTest() {
215
216
217 // Initialize the stats for the MAVs, the team and the target
218 initializeMAVStats();
219 alive = true;
220 target->alive = true;
221 target->damaged = false;
222 target->stepIfDamageIncurred = 0;
223
224 // Assign starting coordinates and angles to the target and to the MAVs
225 target->calculateStartingCoordinates();
226 calculateMAVStartingCoordinates();
227 calculateMAVStartingAngles();
228
229 // Calculate the starting distance between the target and the MAVs
230 calculateTargetMAVDistance();
231
232 // Empty the buffers
233 #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234     emptyMAVBuffers();
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         if (simMAV[currentMAV]->alive) {
249
250 // Calculate MAV's perception
251             calculateMAVPerception();
252
253 // Calculate NN's outputs
254             n_net->NN->step();
255
256 // Fill the buffers
257             #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
258                 fillMAVBuffers();
259             #endif
260
```

Key Idea: Intersect configuration impact with change impact

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

The diagram illustrates the intersection of configuration and change impact. A central code block is surrounded by colored bars and lines. A red line connects the 'pop-up-blocker' option to a red bar. A green line connects the 'Google Toolbar' option to a green bar. A blue line connects the 'Do Not Track' option to a blue bar. An orange bar is also present. The code block contains a function `void Team::performSingleTest()` with various initialization and logic steps. A yellow highlight is placed on a `for` loop that iterates over MAVs. The IE7 logo is visible on the right side of the code block.

```
214 void Team::performSingleTest() {
215
216
217 // Initialize the stats for the MAVs, the team and the target
218 initializeMAVStats();
219 alive = true;
220 target->alive = true;
221 target->damaged = false;
222 target->stepIfDamageIncurred = 0;
223
224 // Assign starting coordinates and angles to the target and to the MAVs
225 target->calculateStartingCoordinates();
226 calculateMAVStartingCoordinates();
227 calculateMAVStartingAngles();
228
229 // Calculate the starting distance between the target and the MAVs
230 calculateTargetMAVDistance();
231
232 // Empty the buffers
233 #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
234     emptyMAVBuffers();
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         if (simMAV[currentMAV]->alive) {
249
250             // Calculate MAV's perception
251             calculateMAVPerception();
252
253             // Calculate NN's outputs
254             n_net->NN->step();
255
256             // Fill the buffers
257             #if ((NN_ARCHITECTURE >= 13) && (NN_ARCHITECTURE <= 24))
258                 fillMAVBuffers();
259             #endif
260
```

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. void f3() { printf("f3"); }
15. void f4() { printf("f4"); }
16. void f5() { printf("f5"); }
17. void f7() { printf("f7"); }
18. void f8() { printf("f8"); }
19.
20. //configurable options
21. unsigned int P1;
22. unsigned int P2;
23. unsigned int P3;
24.
25. void main(){
26.     int x;
27.     if (x == 0) {
28.         if(P1)
29.             f1(1);
30.         else
31.             f2(2);
32.     } // end x==0
33.
34.     else { // x != 0
35.         f3();
36.         if(x < 0){
37.             f5();
38.             if(P3)
39.                 f4();
40.         } // end x < 0
41.
42.         else{ //x > 0
43.             if(P1){
44.                 if(P2)
45.                     f8();
46.                 else
47.                     f7();
48.             }
49.             else
50.                 f6(6);
51.         } // end x > 0
52.     } // end x != 0
53. }
```

Example Program

Mapping configurable options to source code

Configurable Options: $\{P_1, P_2, P_3\}$

<pre>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. void f₃() { printf("f₃"); } 15. void f₄() { printf("f₄"); } 16. void f₅() { printf("f₅"); } 17. void f₇() { printf("f₇"); } 18. void f₈() { printf("f₈"); } 19. 20. //configurable options 21. unsigned int P₁; 22. unsigned int P₂; 23. unsigned int P₃;</pre>	<pre>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. } // end x != 0 57. } 58. }</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Example Program

Mapping configurable options to source code

Configurable Options: $\{P_1, P_2, P_3\}$

<pre>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. void f₃() { printf("f₃"); } 15. void f₄() { printf("f₄"); } 16. void f₅() { printf("f₅"); } 17. void f₇() { printf("f₇"); } 18. void f₈() { printf("f₈"); } 19. //configurable options 20. unsigned int P₁; 21. unsigned int P₂; 22. unsigned int P₃;</pre>	<pre>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. } // end x != 0 57. } 58. }</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Example Program

Mapping configurable options to source code

Configurable Options: $\{P_1, P_2, P_3\}$

<pre>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. void f₃() { printf("f₃"); } 15. void f₄() { printf("f₄"); } 16. void f₅() { printf("f₅"); } 17. void f₇() { printf("f₇"); } 18. void f₈() { printf("f₈"); } 19. //configurable options 20. unsigned int P₁; 21. unsigned int P₂; 22. unsigned int P₃;</pre>	<pre>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. } // end x != 0 57. } 58. }</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Example Program

Mapping configurable options to source code

Configurable Options: $\{P_1, P_2, P_3\}$

Function f_1 changes

<pre>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. void f₃() { printf("f₃"); } 15. void f₄() { printf("f₄"); } 16. void f₅() { printf("f₅"); } 17. void f₇() { printf("f₇"); } 18. void f₈() { printf("f₈"); } 19. 20. //configurable options 21. unsigned int P₁; 22. unsigned int P₂; 23. unsigned int P₃;</pre>	<pre>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. } // end x != 0 57. } 58. }</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Example

Example

Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Example

Configurable Options

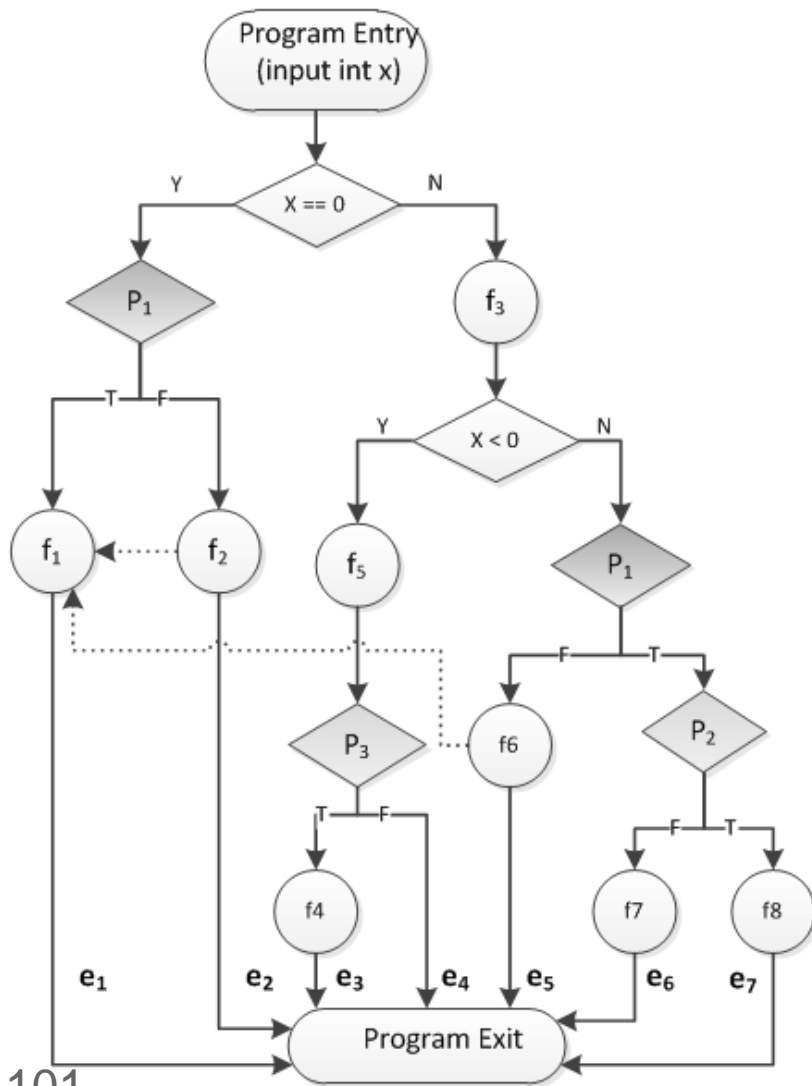
Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

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

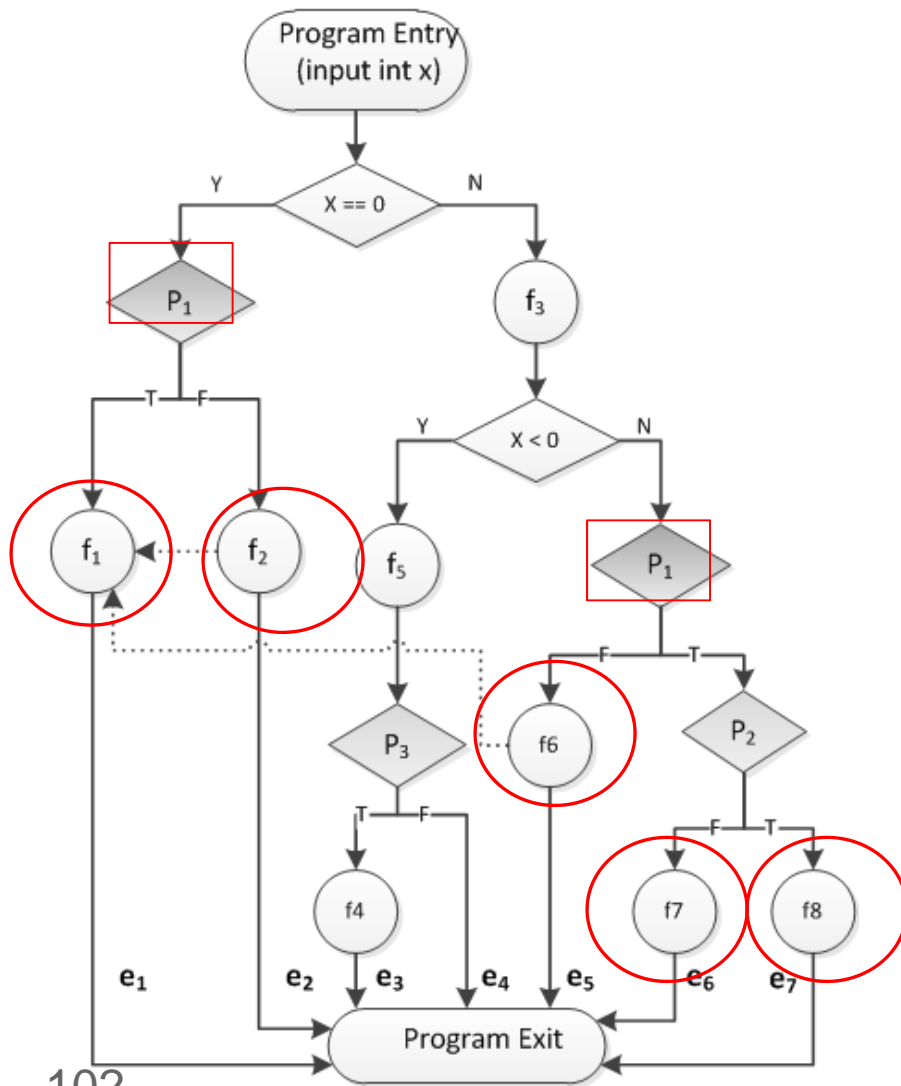
Impact of configuration option P_1 $f_1, f_2, f_6, f_7,$ and f_8

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



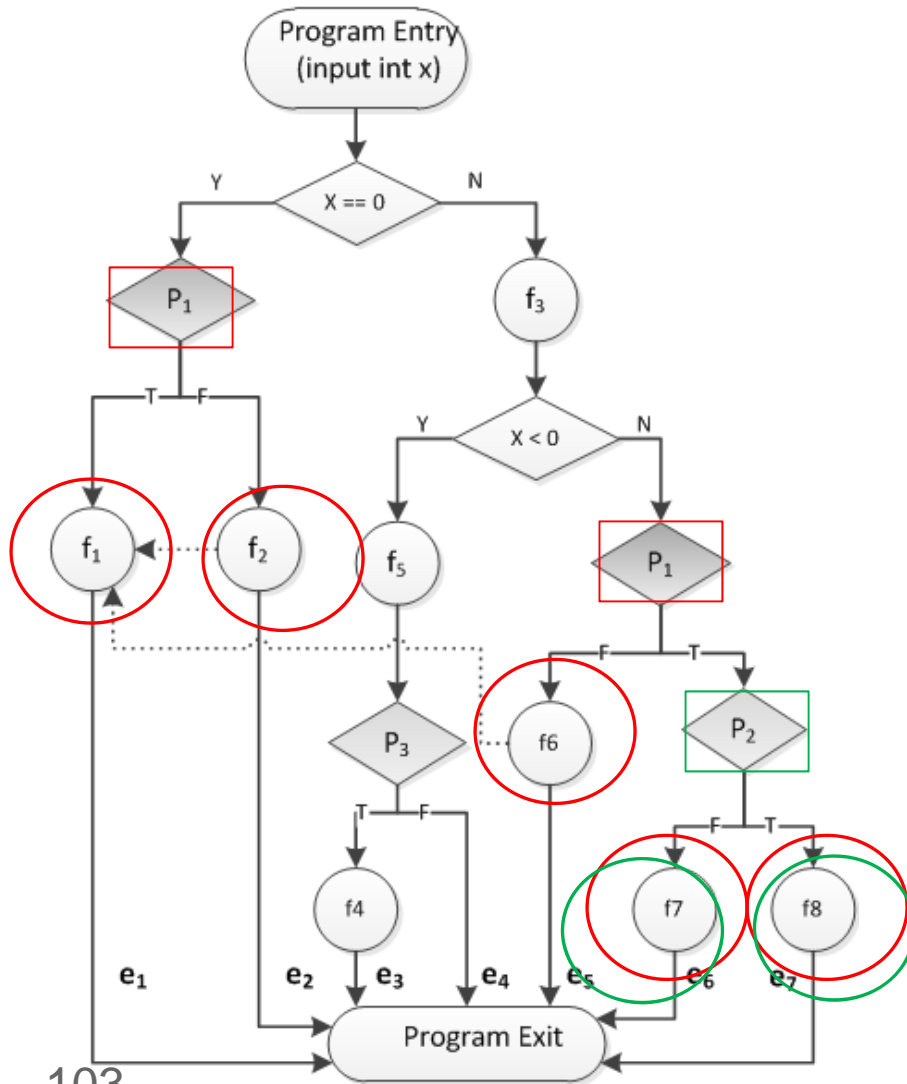
Impact of configuration option P_2 f_7 and f_8

Configurable Options

Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

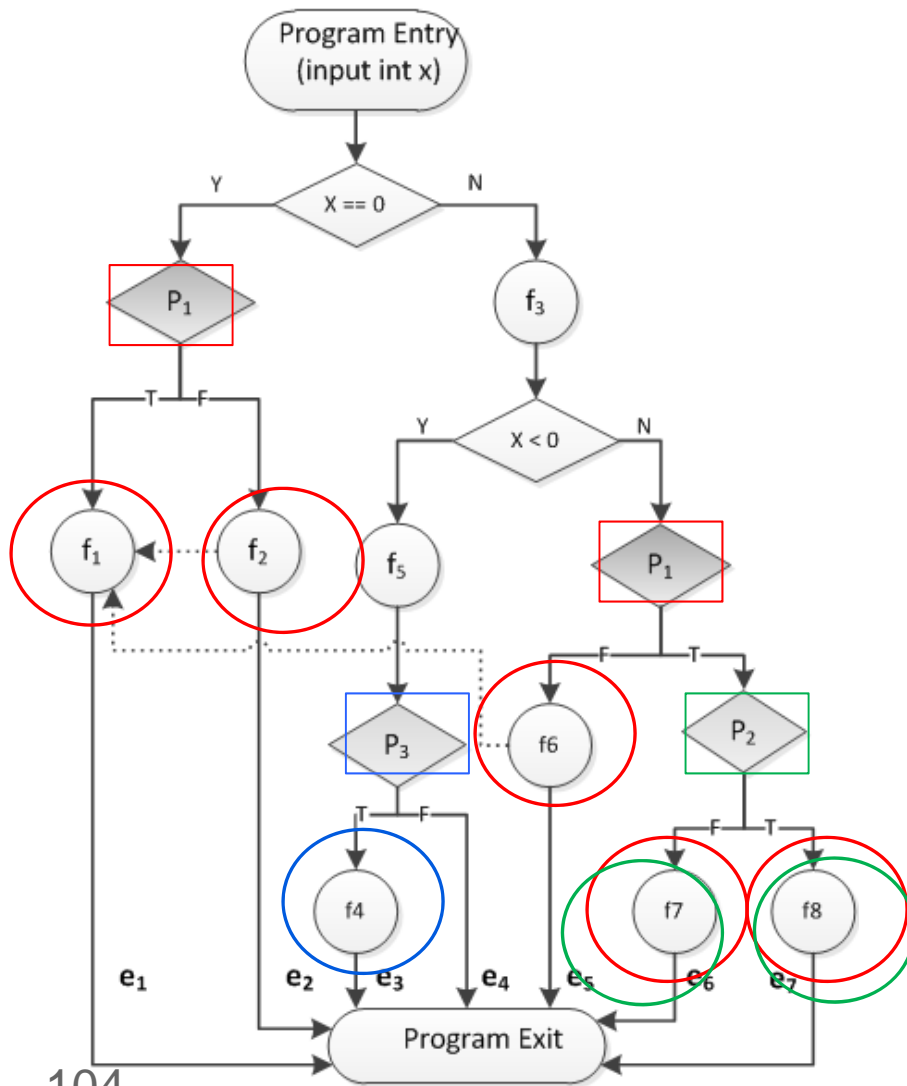
Configurations by pair-wise CIT

	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



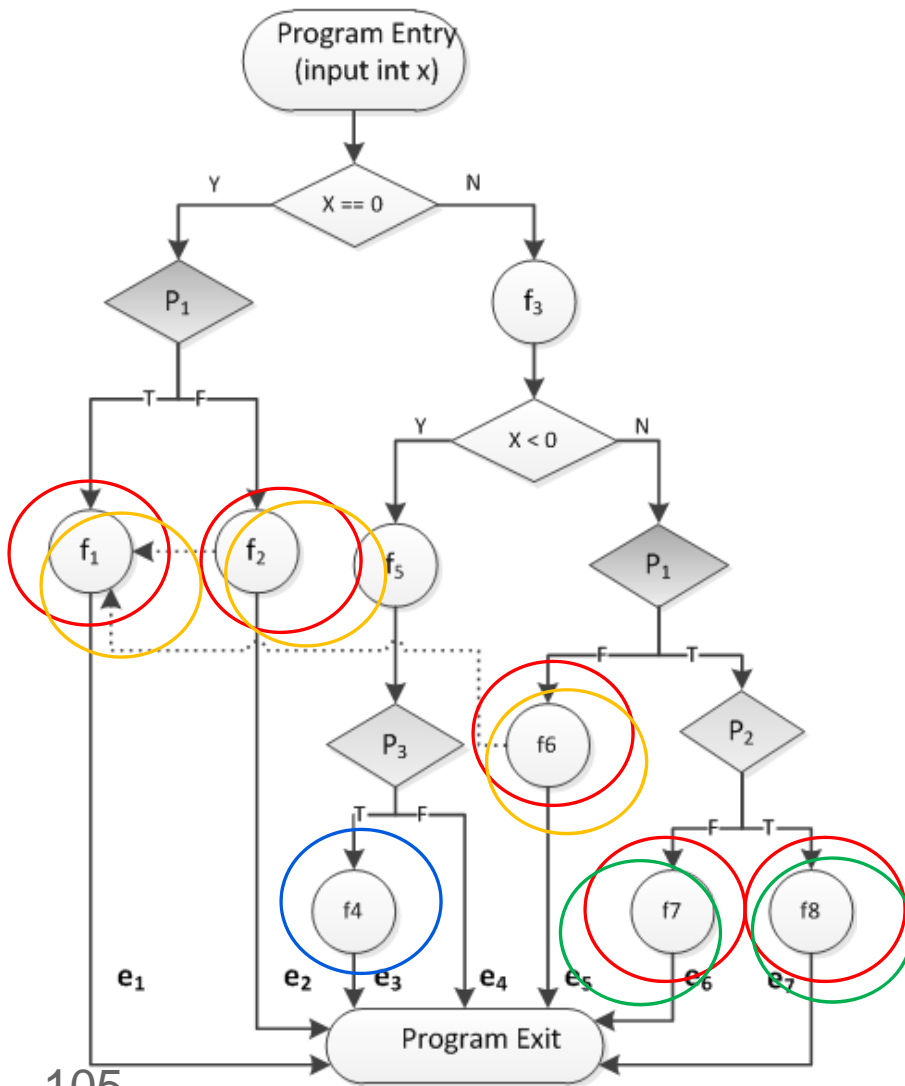
Configurable Options

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

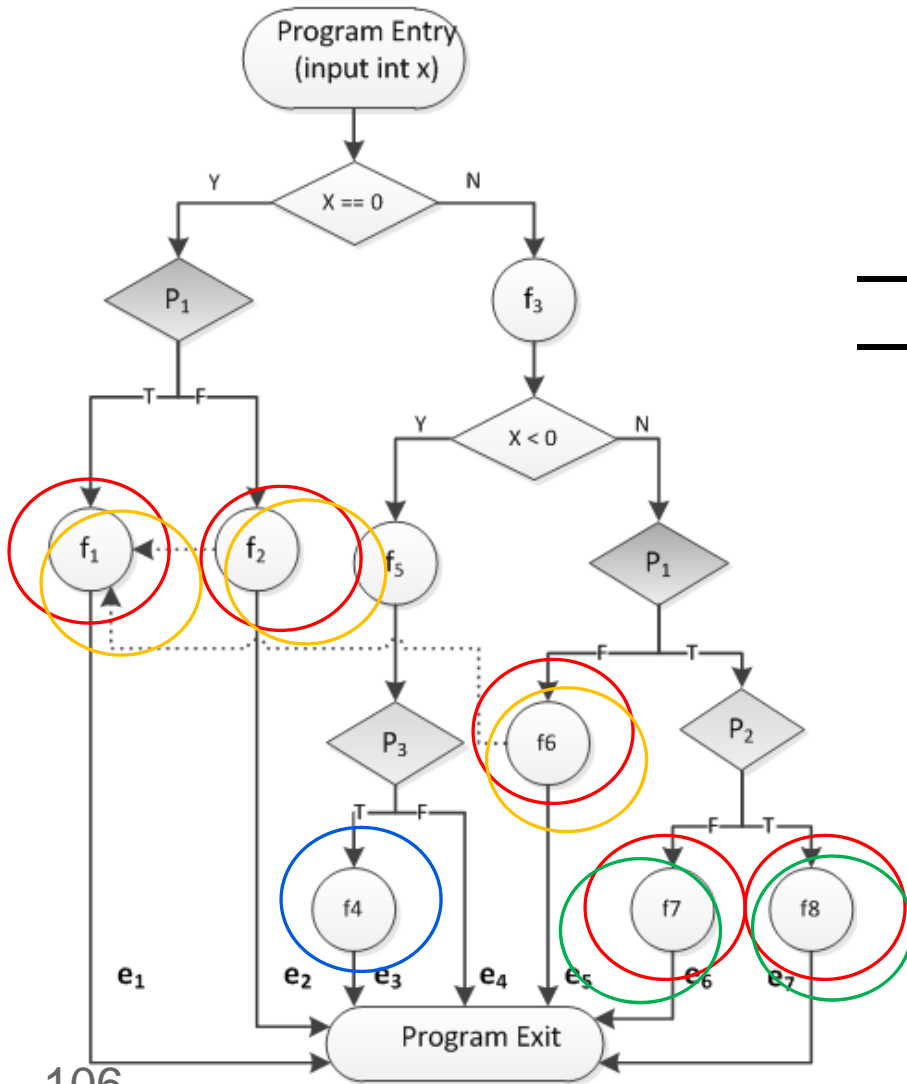
Select option P_1 and safely discard P_2 and P_3

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



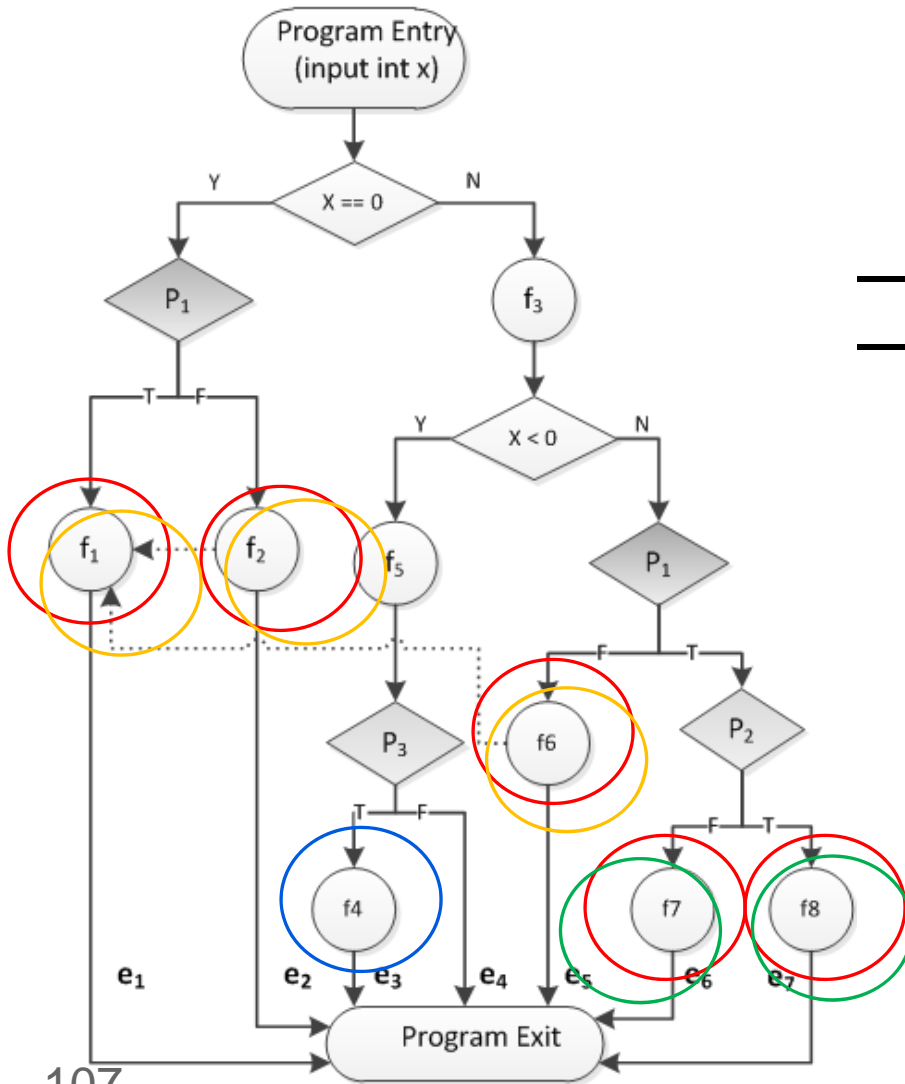
Select option P_1 and safely discard P_2 and P_3

Configurable Options

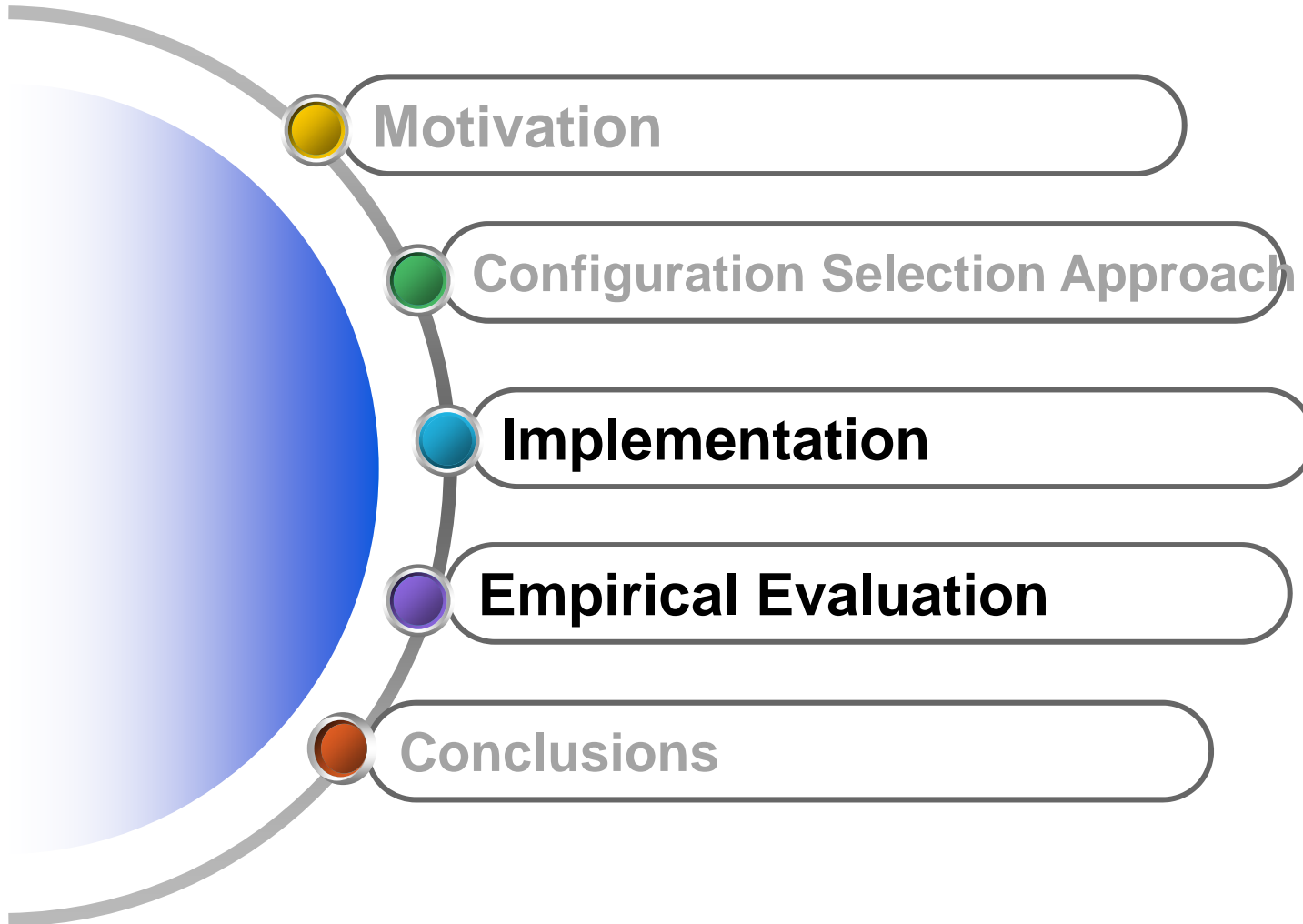
Options	Values	
P_1	True	False
P_2	True	False
P_3	True	False

Configurations by pair-wise CIT

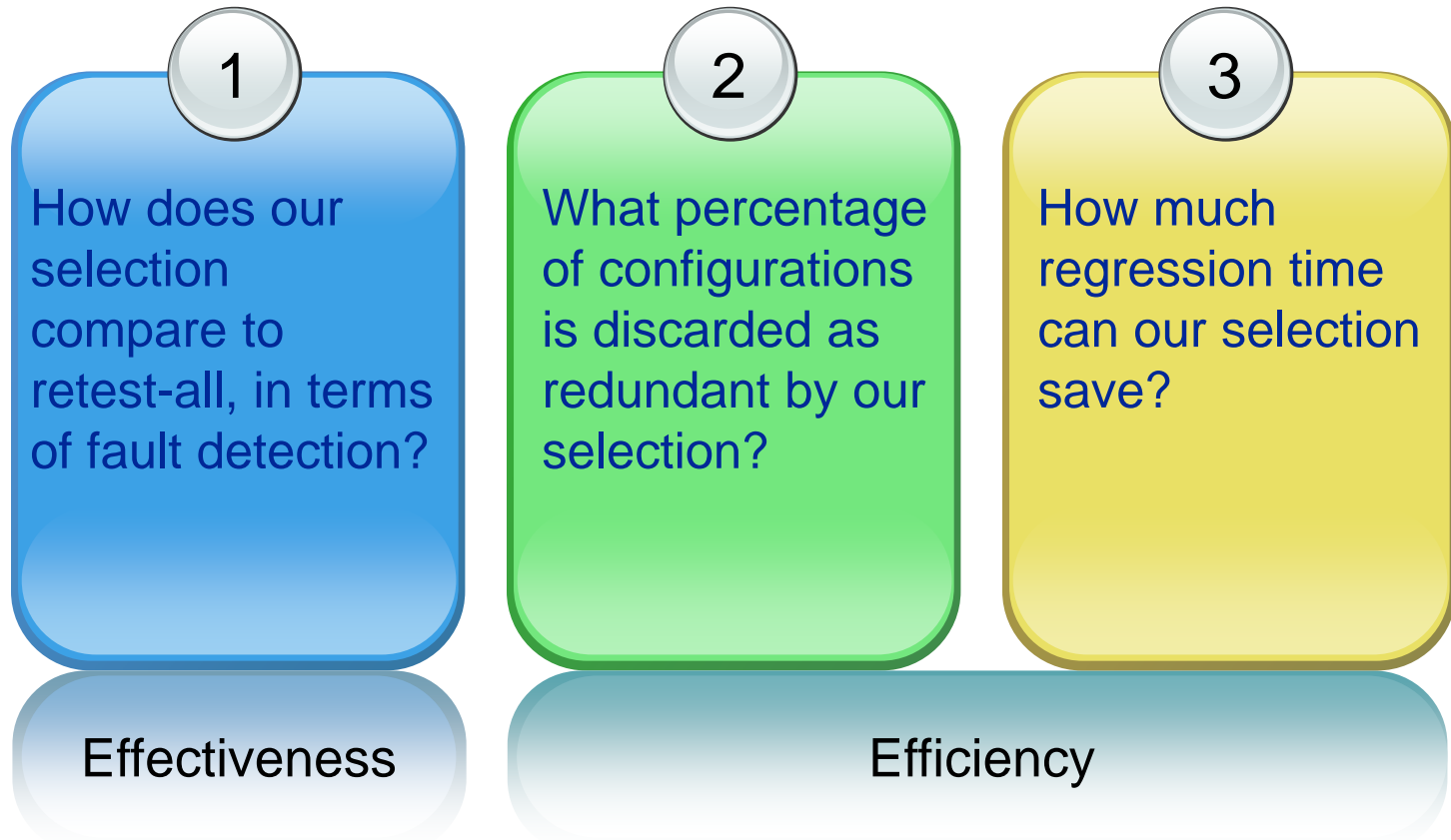
	P_1	P_2	P_3
C_1	True	True	True
C_2	True	False	False
C_3	False	True	False
C_4	False	False	True



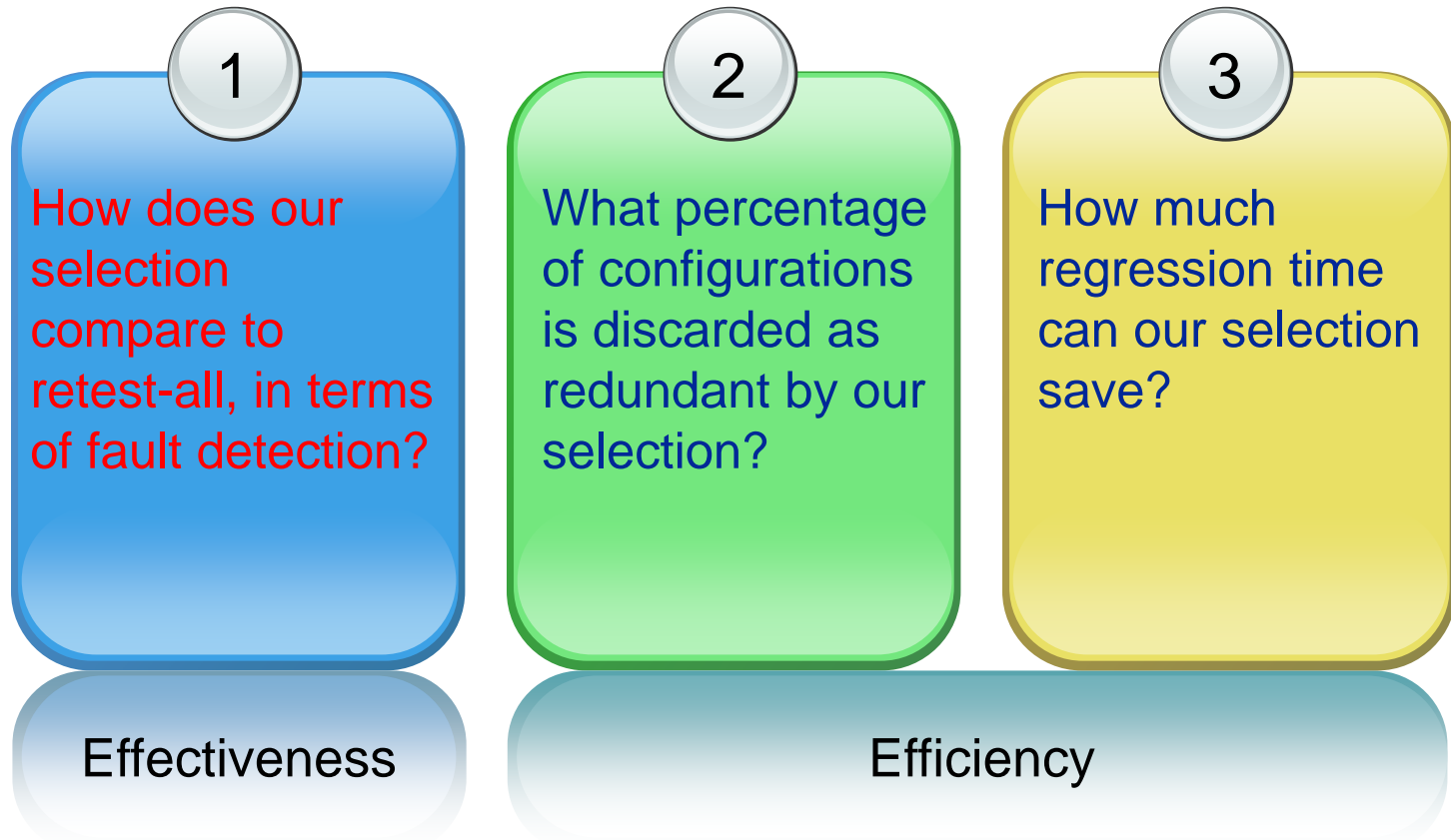
Outline



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

	<i>Make</i>	<i>Grep</i>
Retest-all	8/15	6/15
Our selection	8/15	6/15
Random selection	3/15	5/15

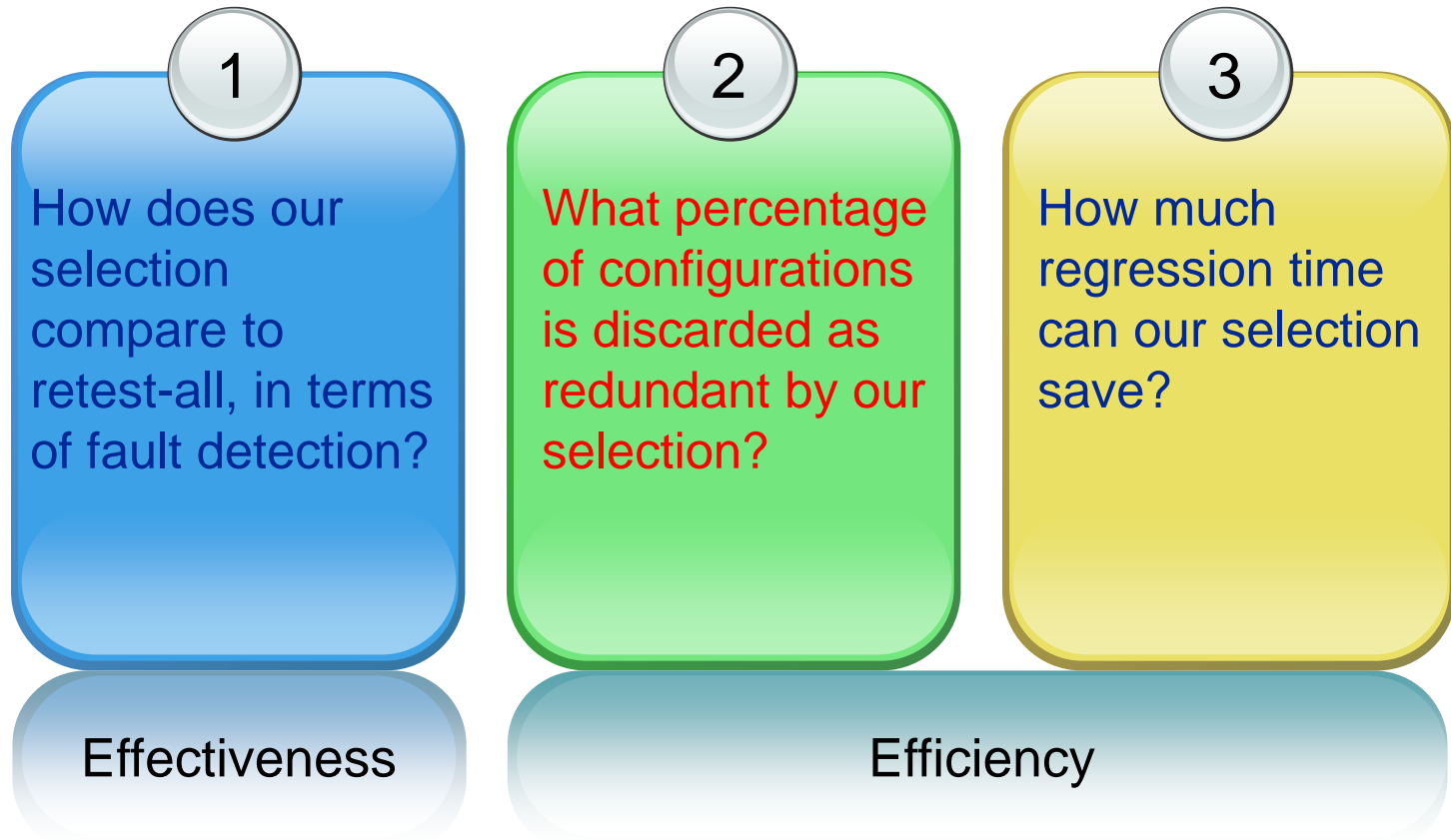
Results

Fault Detection Ability

	<i>Make</i>	<i>Grep</i>
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

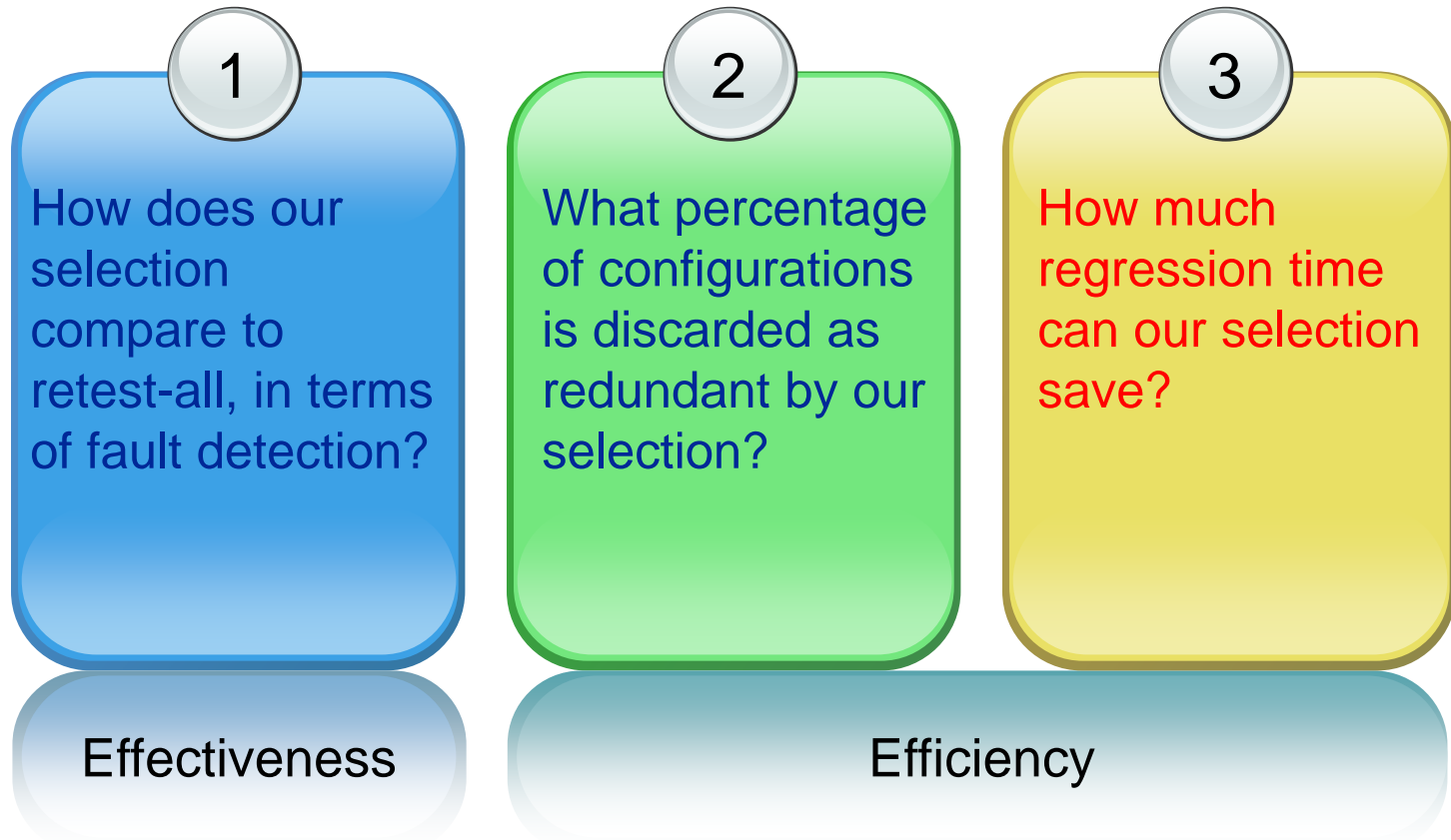
NUMBER OF CONFIGURABLE OPTIONS SELECTED

	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%

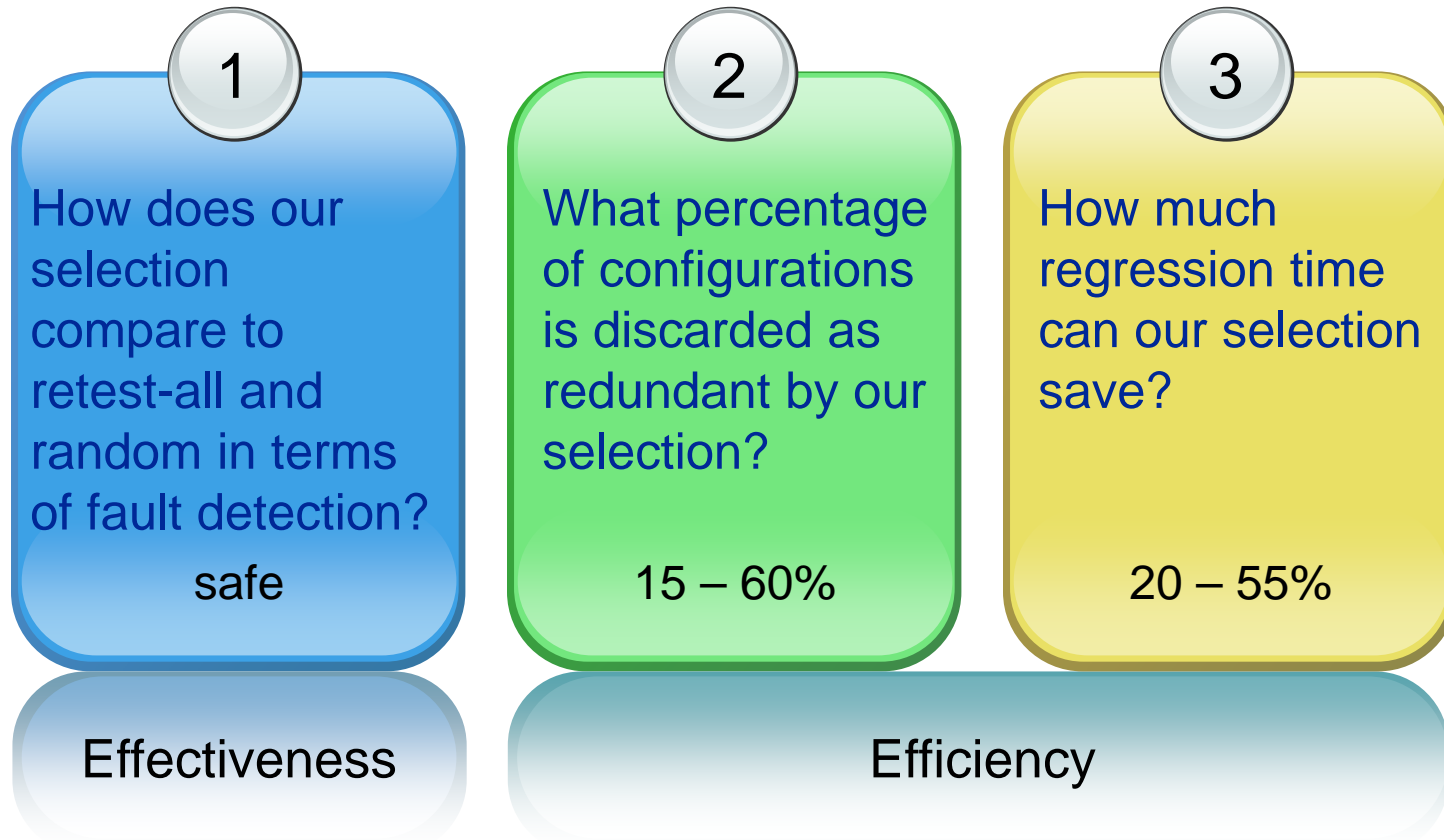
Results

Testing time savings

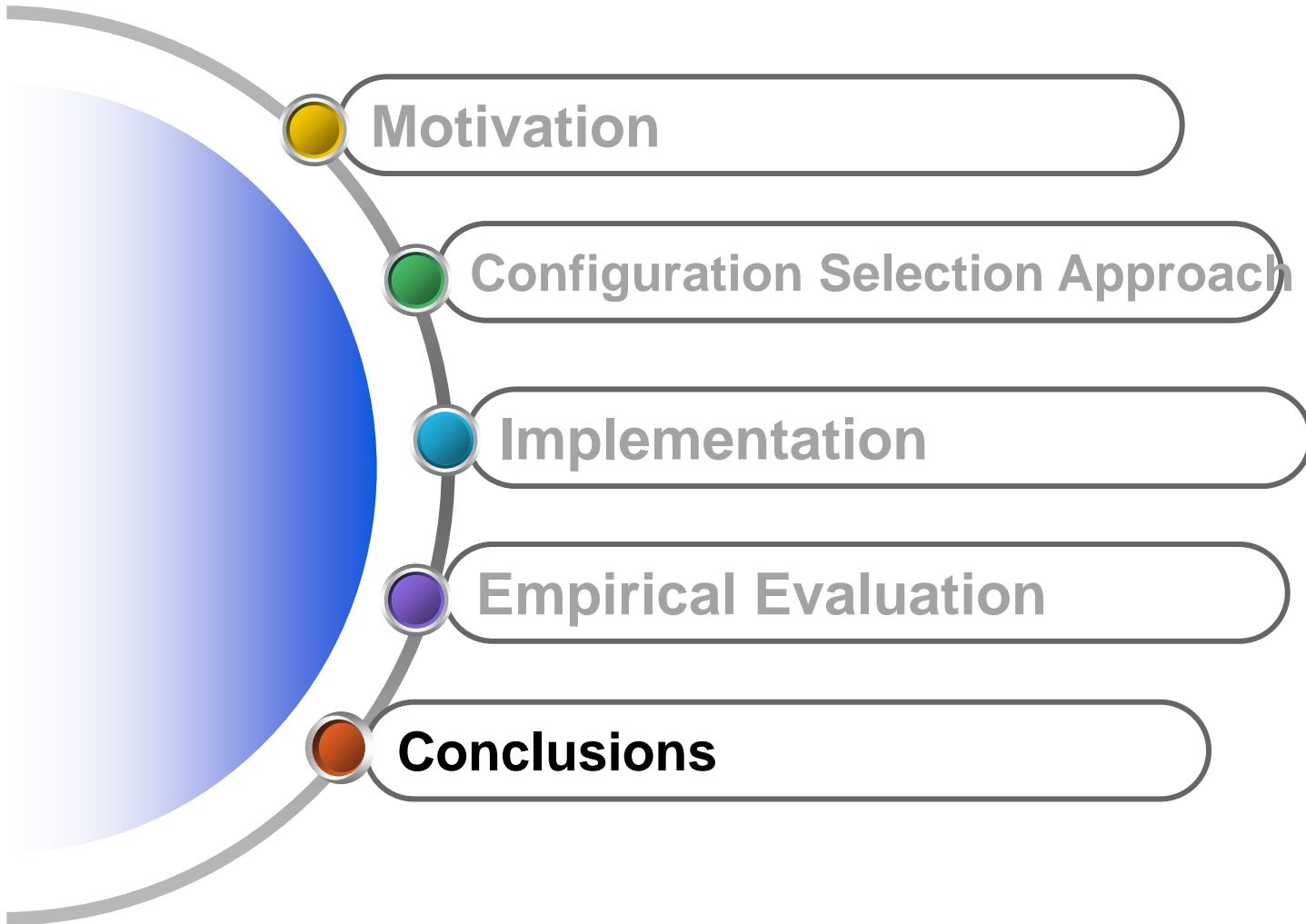
		grep	make	ABB1
Testing time	Retest-all	70m	700m	795h
	Our approach	60m	300m	600h
Overhead of selection		5.2m	13m	28h
Time savings		5m	387m	167h
		50%	55%	21%

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

Better than random, safe wrt retest-all



Outline



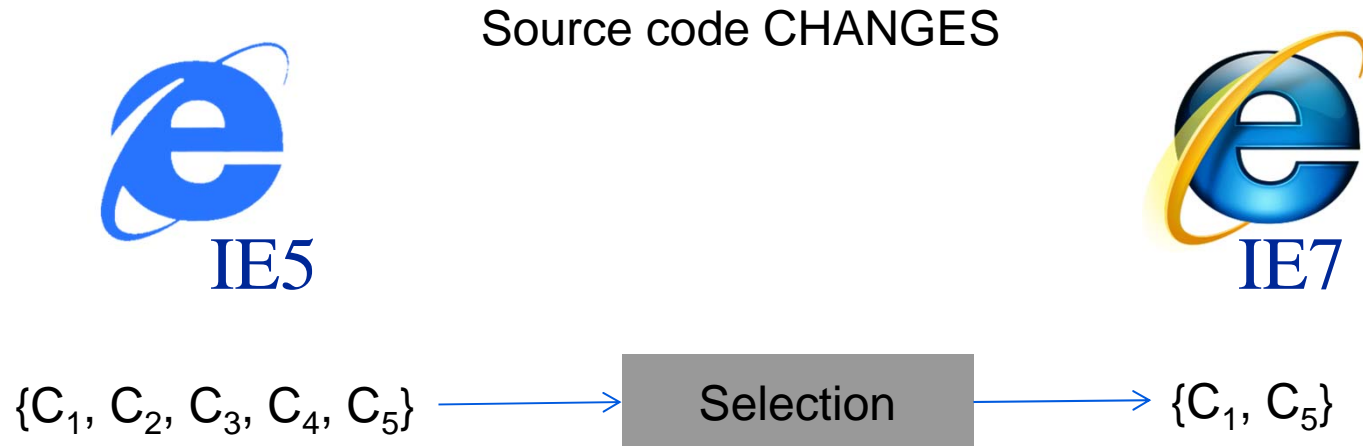
First configuration selection approach for regression testing

First configuration selection approach for regression testing

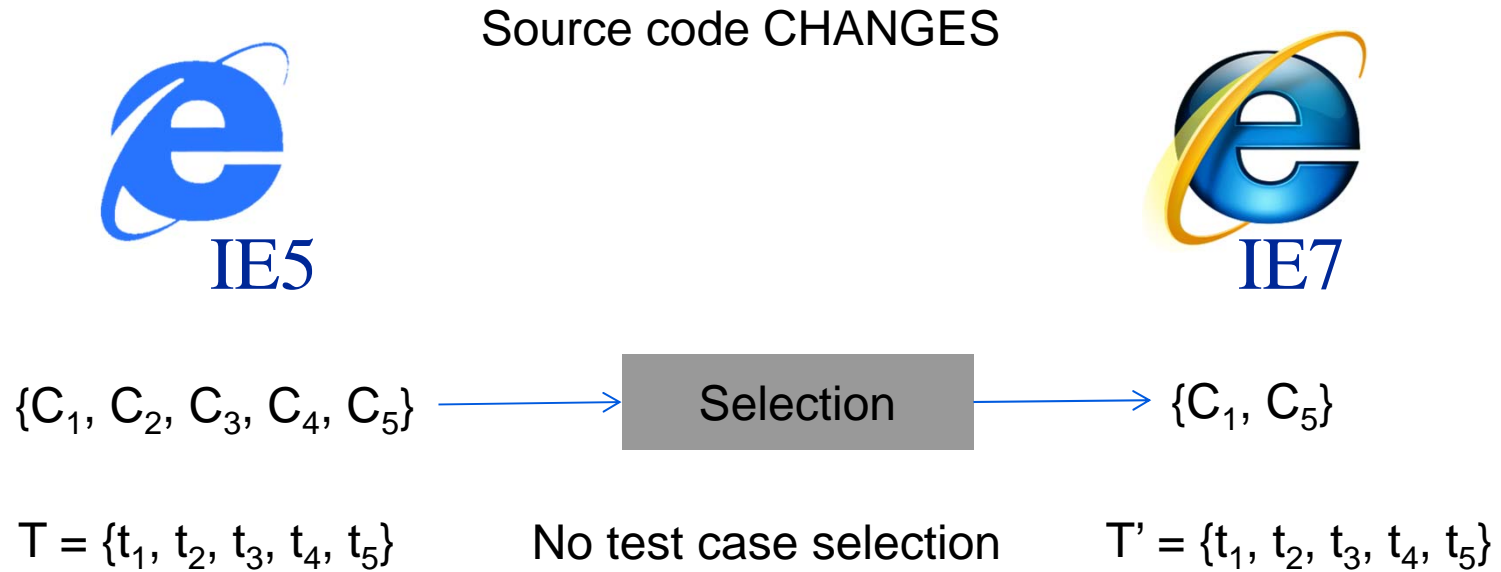
Source code CHANGES



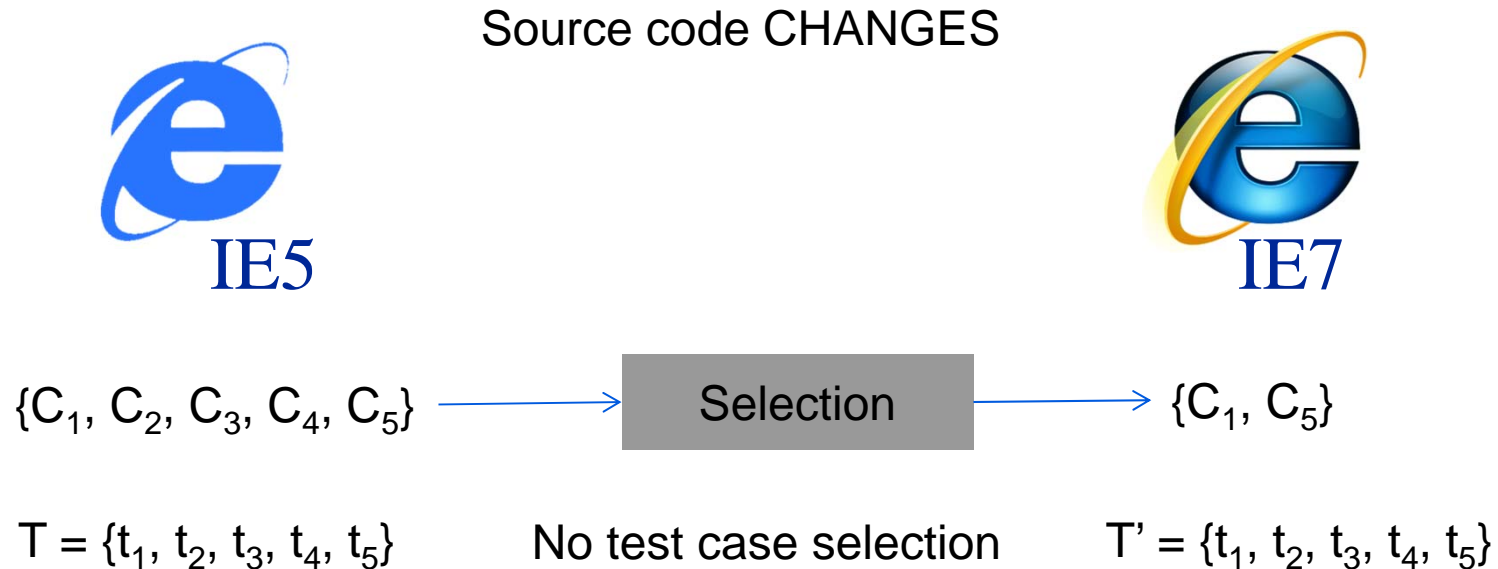
First configuration selection approach for regression testing



First configuration selection approach for regression testing

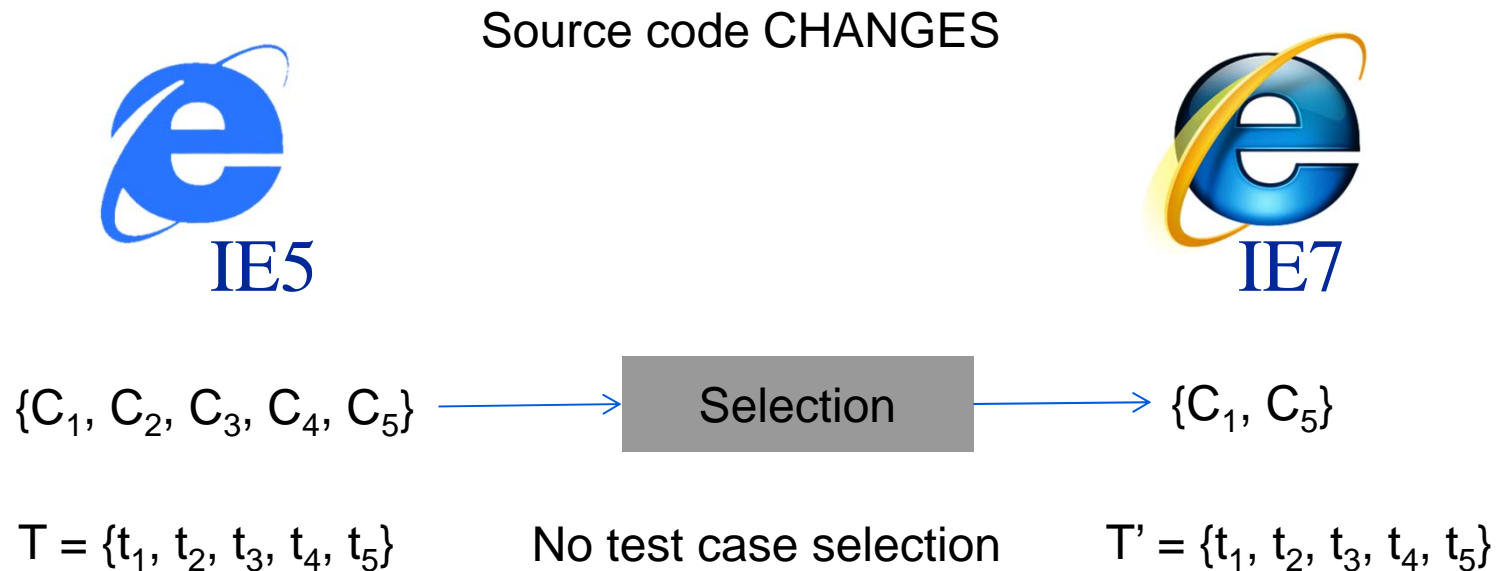


First configuration selection approach for regression testing



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

First configuration selection approach for regression testing



$\{C_1, C_5\}$ 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™

