Using visual models to boost your test automation

Tomas Rosenqvist & Johan Rönnlund

AFA Försäkring

Who are we?

QA technical/automation leads

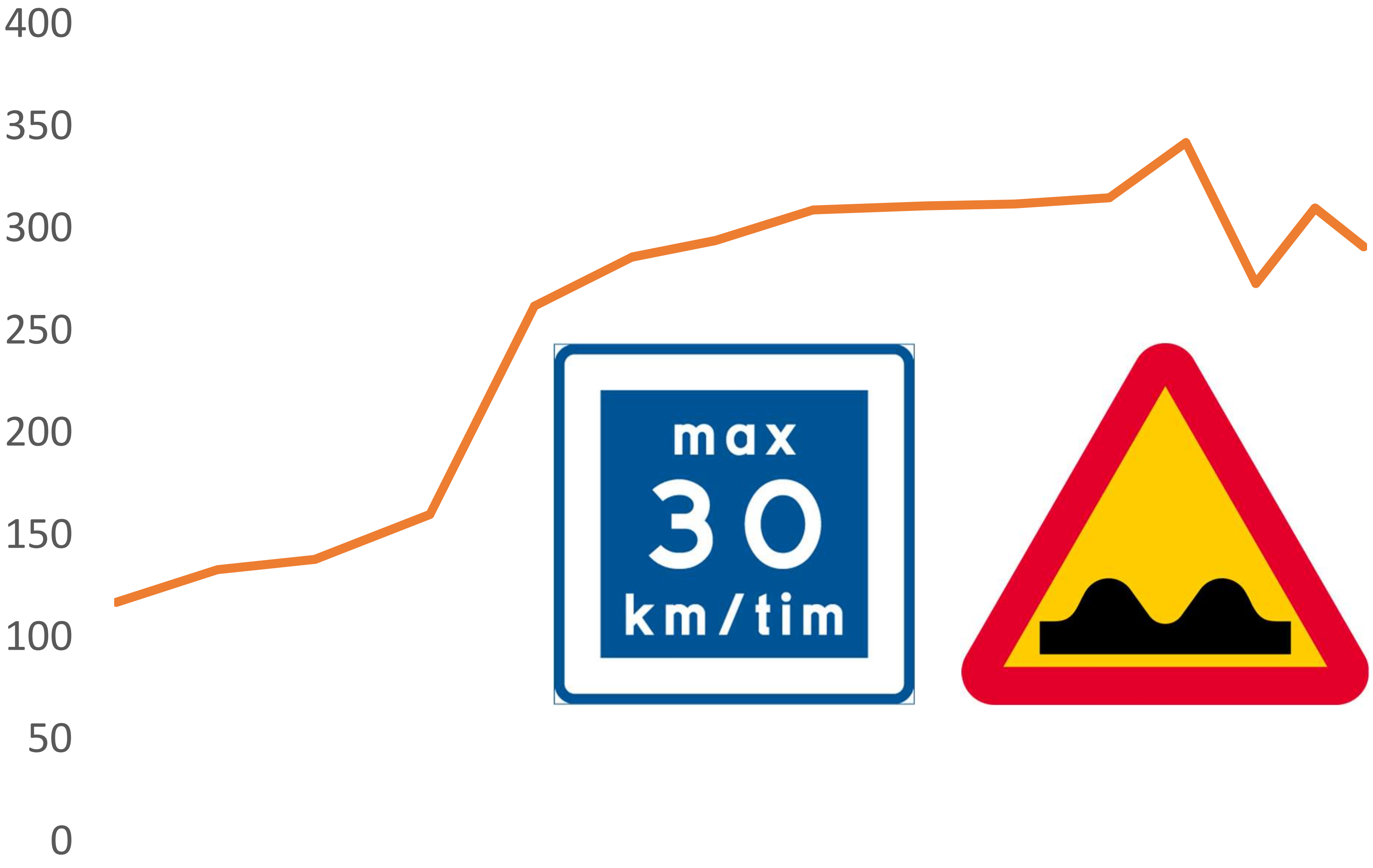
- .NET development
- CI/CD
- API design
- QA evangelism

(basically .NET developers with a QA touch)

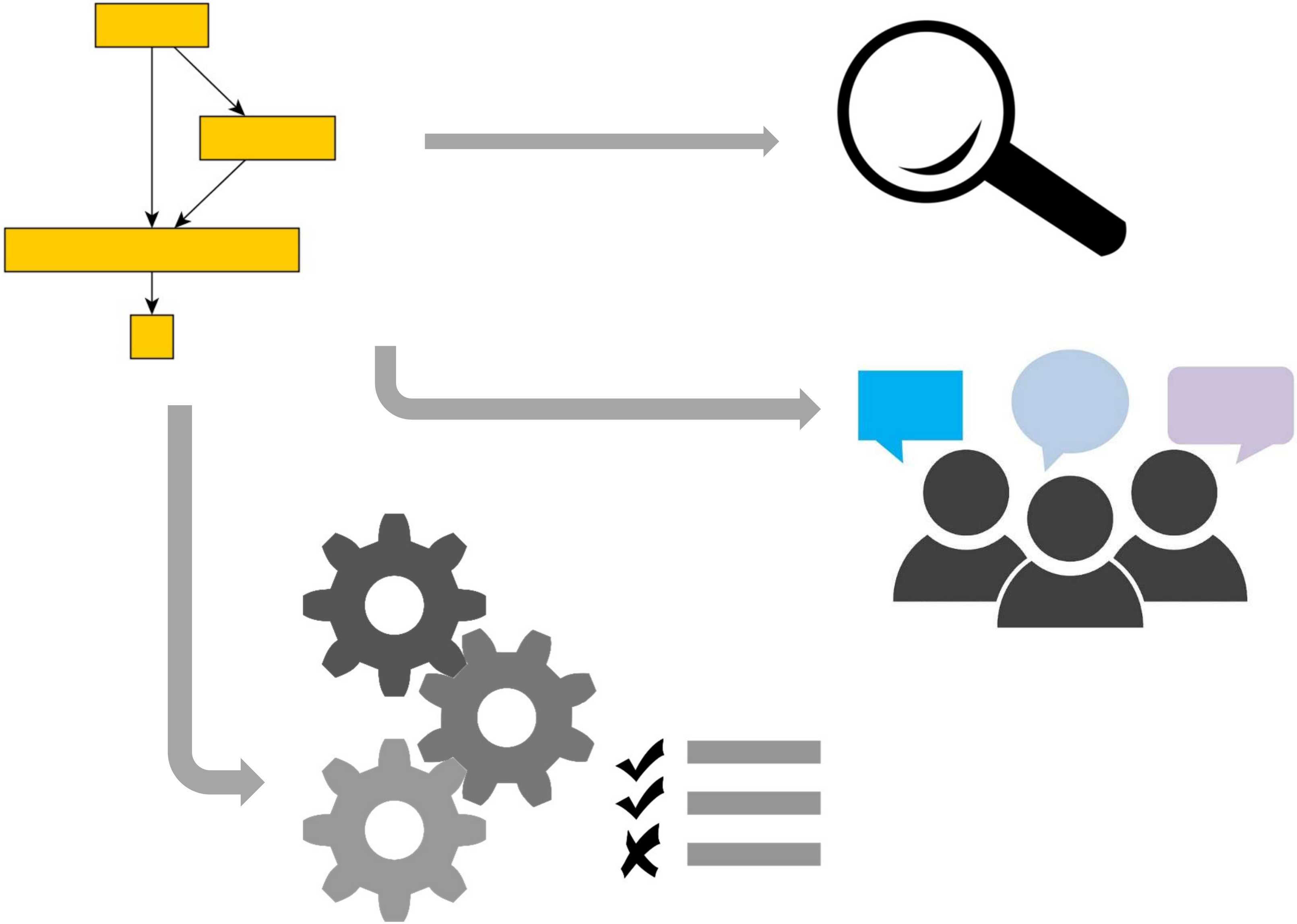
What is AFA Försäkring?

- The Swedish model
  - Employee safety for over 4.5 million adults, but…
  - most of them don’t realize they are covered.
  - Claims are typically reported on the web by the individuals themselves.
- Roughly 200 coworkers @ IT department
  - Software development using Microsoft stack and Progress OpenEdge.

Our test automation effort



Why did we go for MBT?



What is model-based testing?

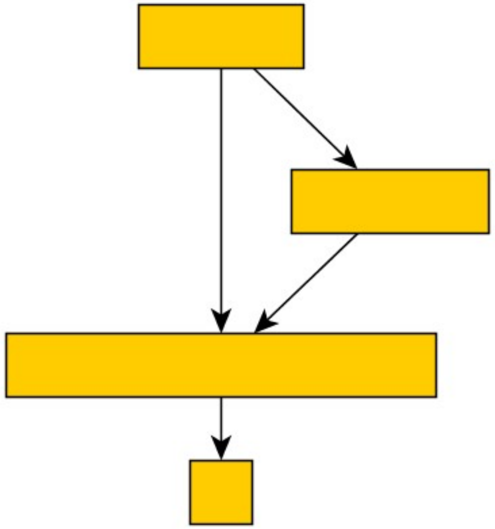
*Models are the **expected behavior** of a System Under Test.*

*Models are **much simpler** than the reality.*

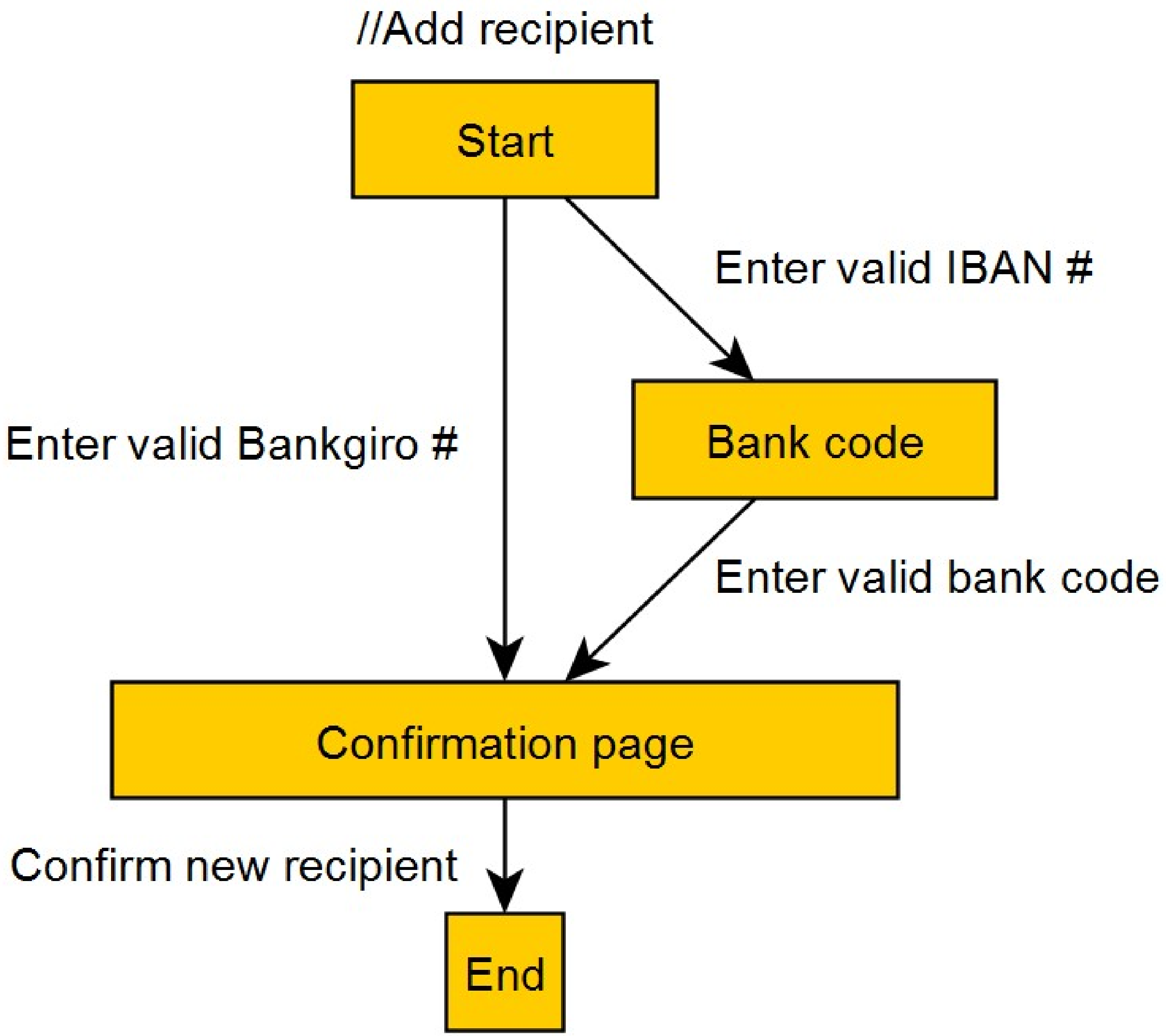
(Kristian Karl, graphwalker.org)

There can be several **different** models for the same System Under Test.

Modeling 101



A very simple model (online bank)

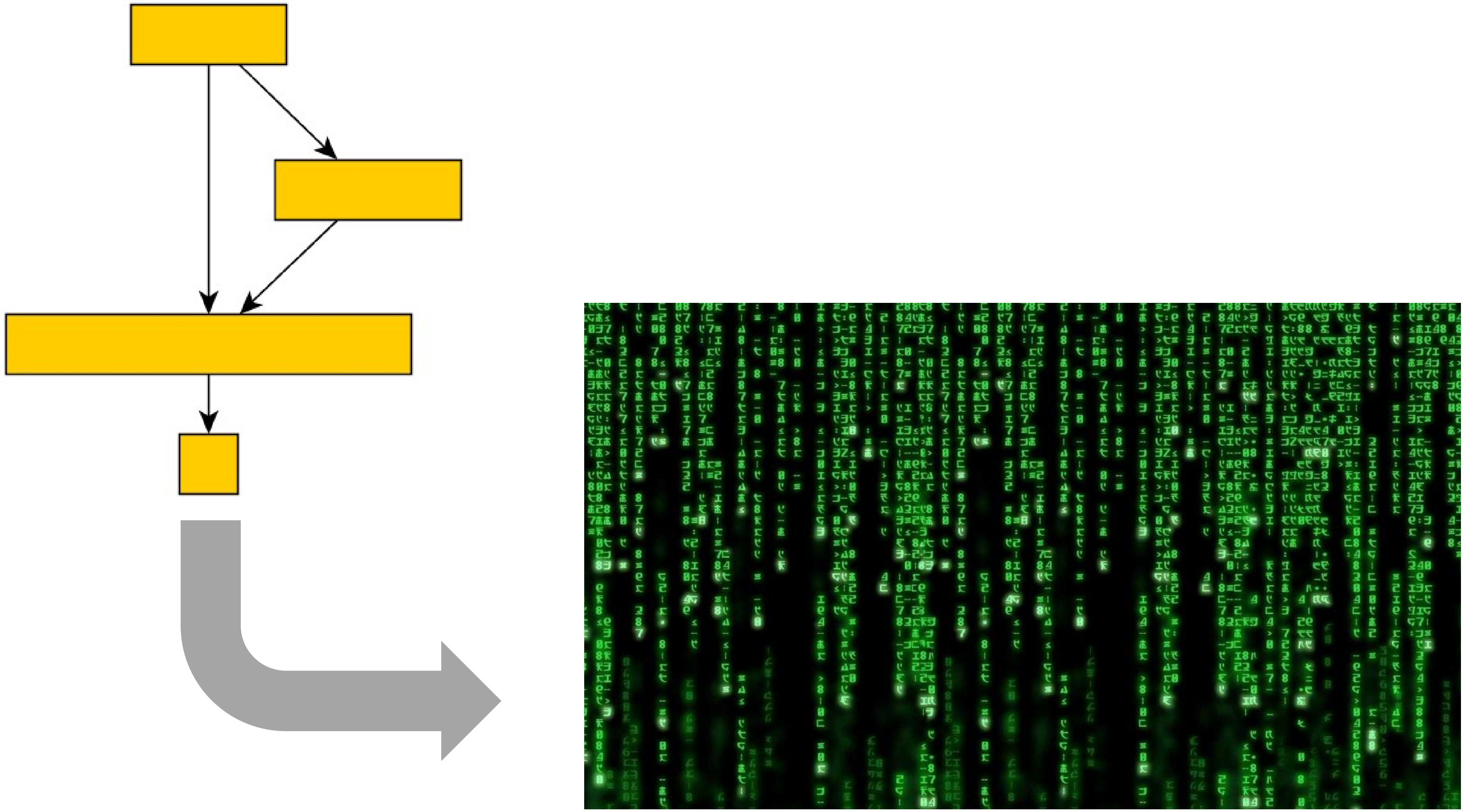


Benefits of MBT

MBT tool survey

- Graphwalker
- CA Agile Requirements Designer

Translating a model into code

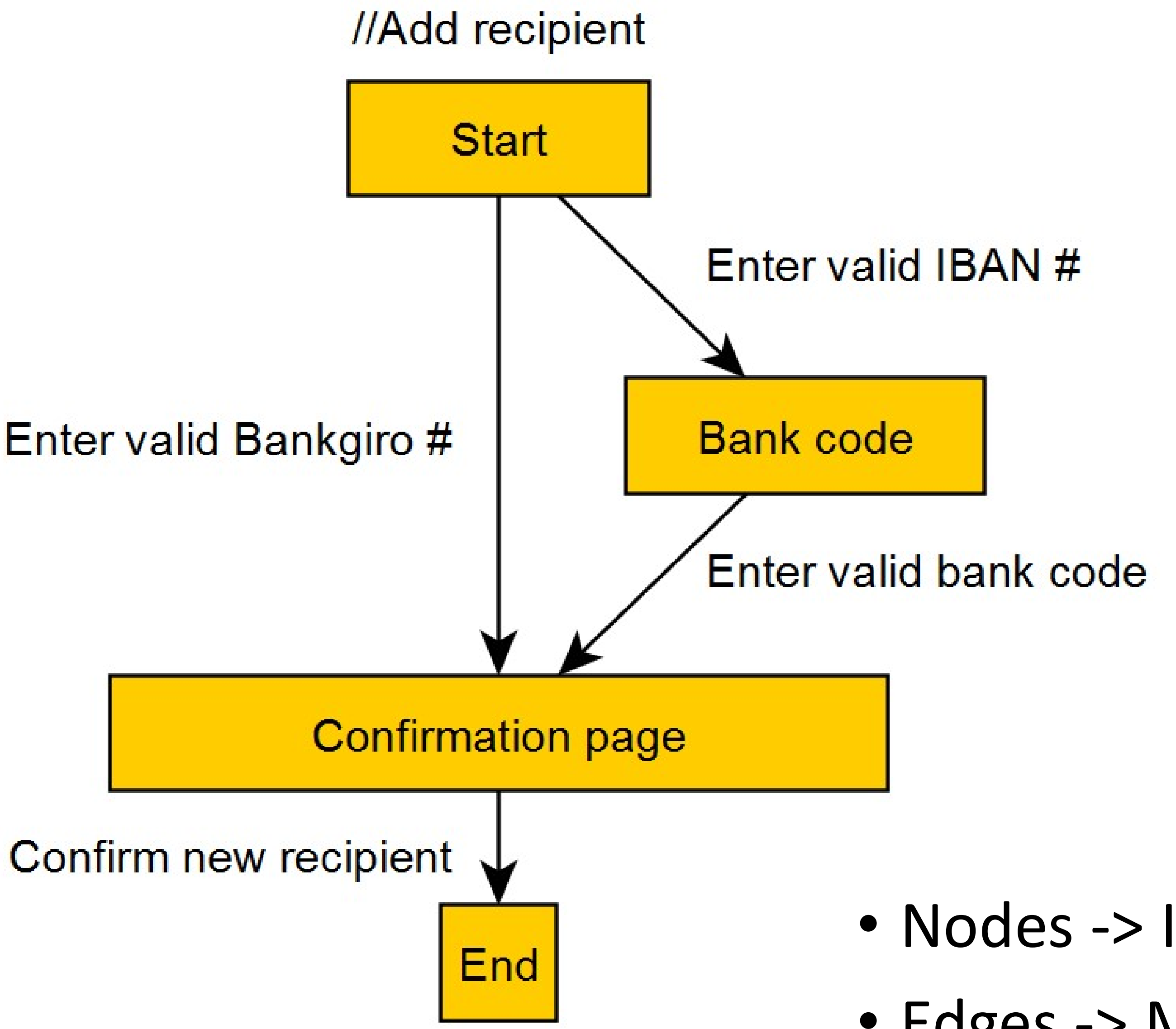


What kind of code do we want?

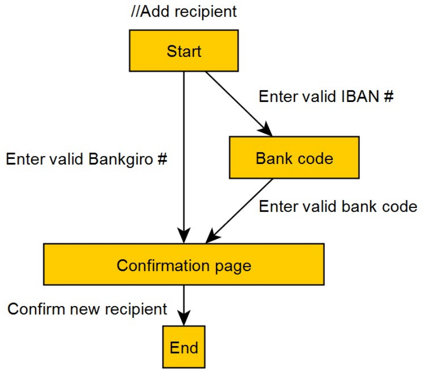
- Understandable (to us)
- Immediately compilable
- Defines behaviors
- Usable for humans
- Shared code

C#

Interfaces

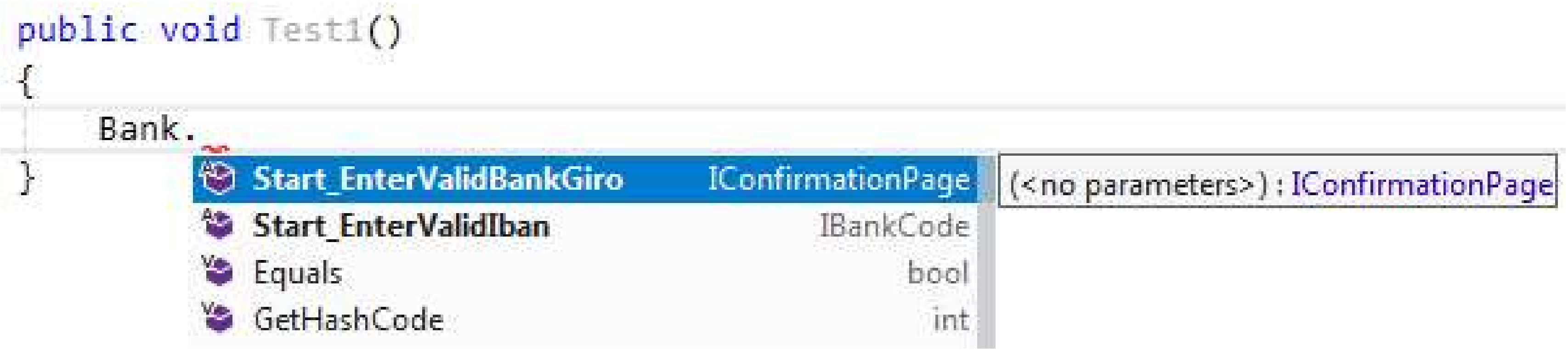


- Nodes -> Interfaces
- Edges -> Methods



```
interface IStart
{
    IBankCode Start_EnterValidIban();
    IConfirmationPage Start_EnterValidBankGiro();
}
interface IBankCode
{
    IConfirmationPage BankCode_EnterValidBankCode();
}
interface IConfirmationPage
{
    IEnd ConfirmationPage_ConfirmNewRecipient();
}
interface IEnd { }
```

Our first test case

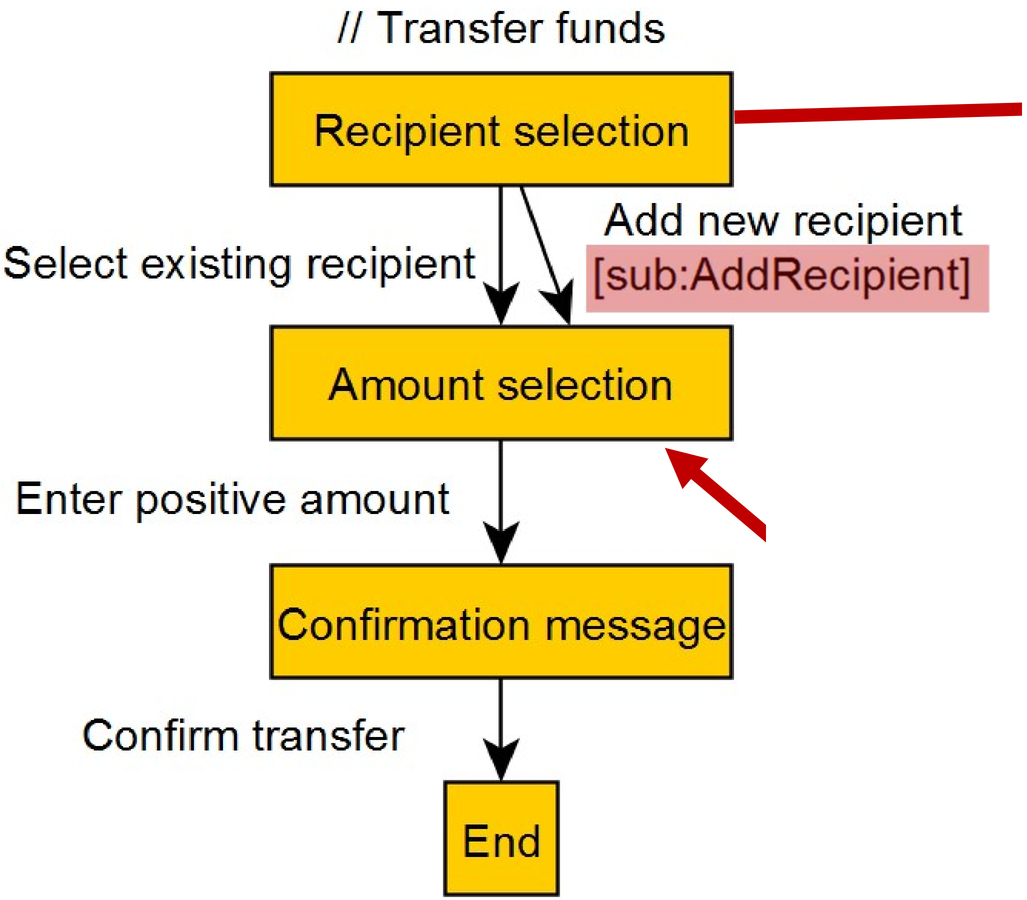


```
public void Test1()
{
    Bank.
}
```

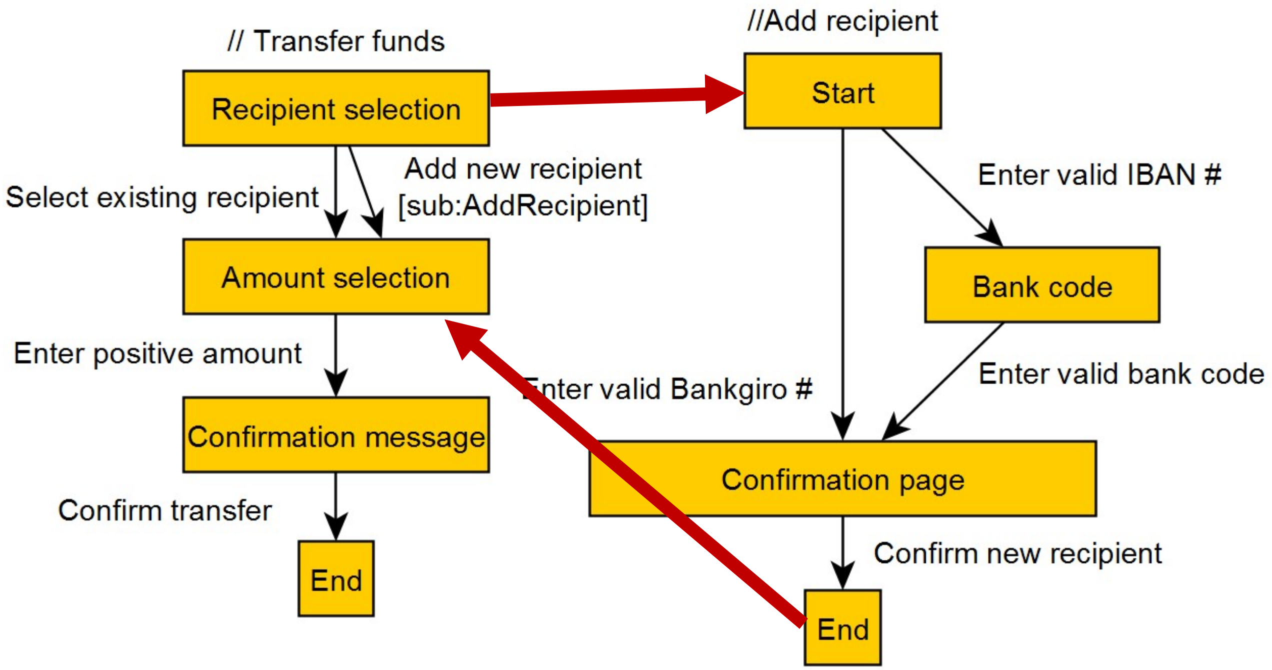
Start_EnterValidBankGiro IConfirmationPage
Start_EnterValidIban IBankCode
Equals bool
GetHashCode int
(<no parameters>) : IConfirmationPage

```
public void Test()
{
    Bank.Start_EnterValidIban()
        .BankCode_EnterValidBankCode()
        .ConfirmationPage_ConfirmNewRecipient();
}
```

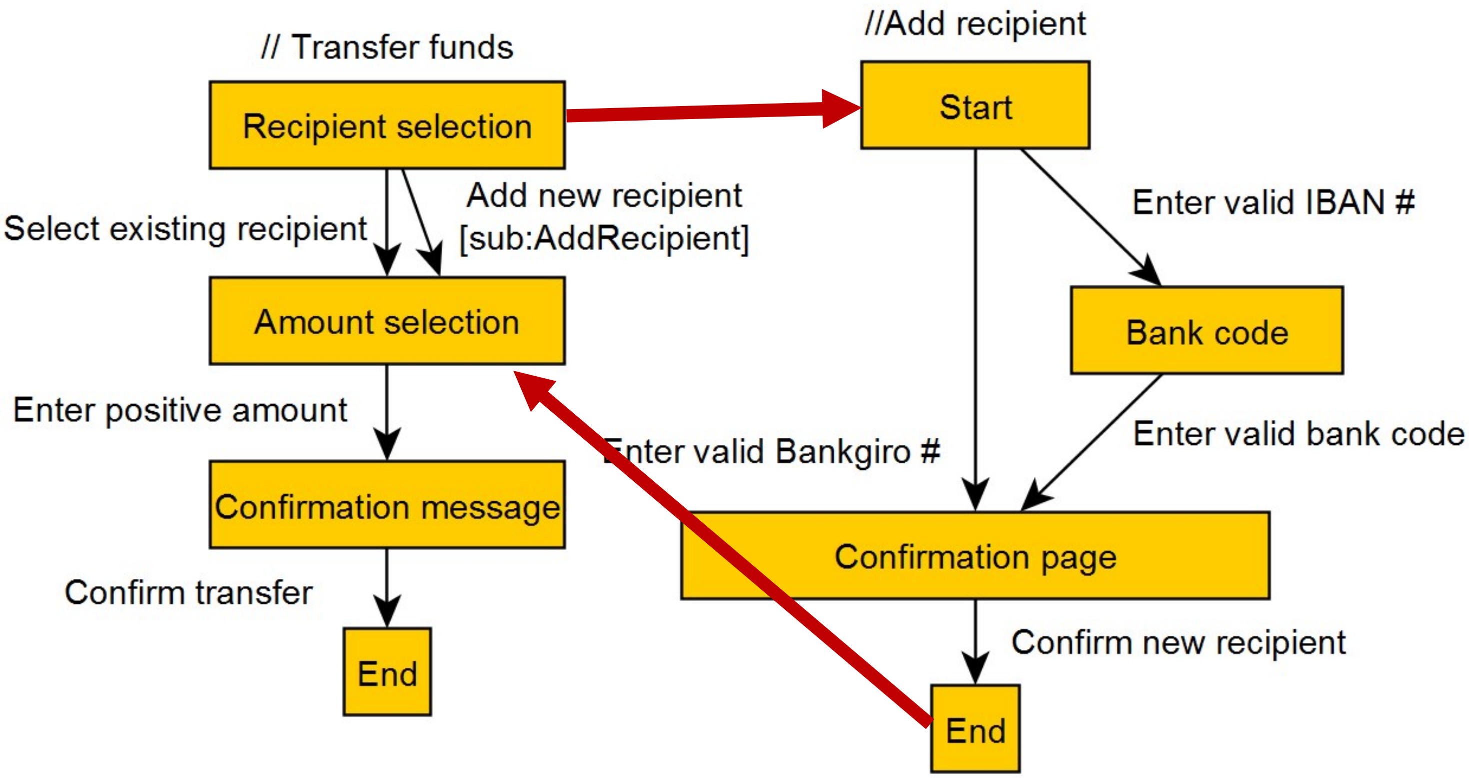
Dividing & re-using models



```
public interface IRecipientSelection
{
    IAmountSelection SelectExistingRecipient();
    IAmountSelection AddNewRecipient(Func<IStart, IEnd> func);
}
public interface IAmountSelection
{
    IConfirmationMessage EnterPositiveAmount();
}
public interface IConfirmationMessage
{
    IEnd ConfirmTransfer();
}
```



```
public void Test2()
{
    Bank.AddNewRecipient(addRecipient =>
            addRecipient.Start_EnterValidIban()
                .BankCode_EnterValidBankCode()
                .ConfirmationPage_ConfirmNewRecipient())
        .EnterPositiveAmount()
        .ConfirmTransfer();
}
```



// Transfer funds

Recipient selection

Select existing recipient

Add new recipient [sub:AddRecipient]

Amount selection

Enter positive amount

Confirmation message

Confirm transfer

End

//Add recipient

Start

Enter valid IBAN #

Bank code

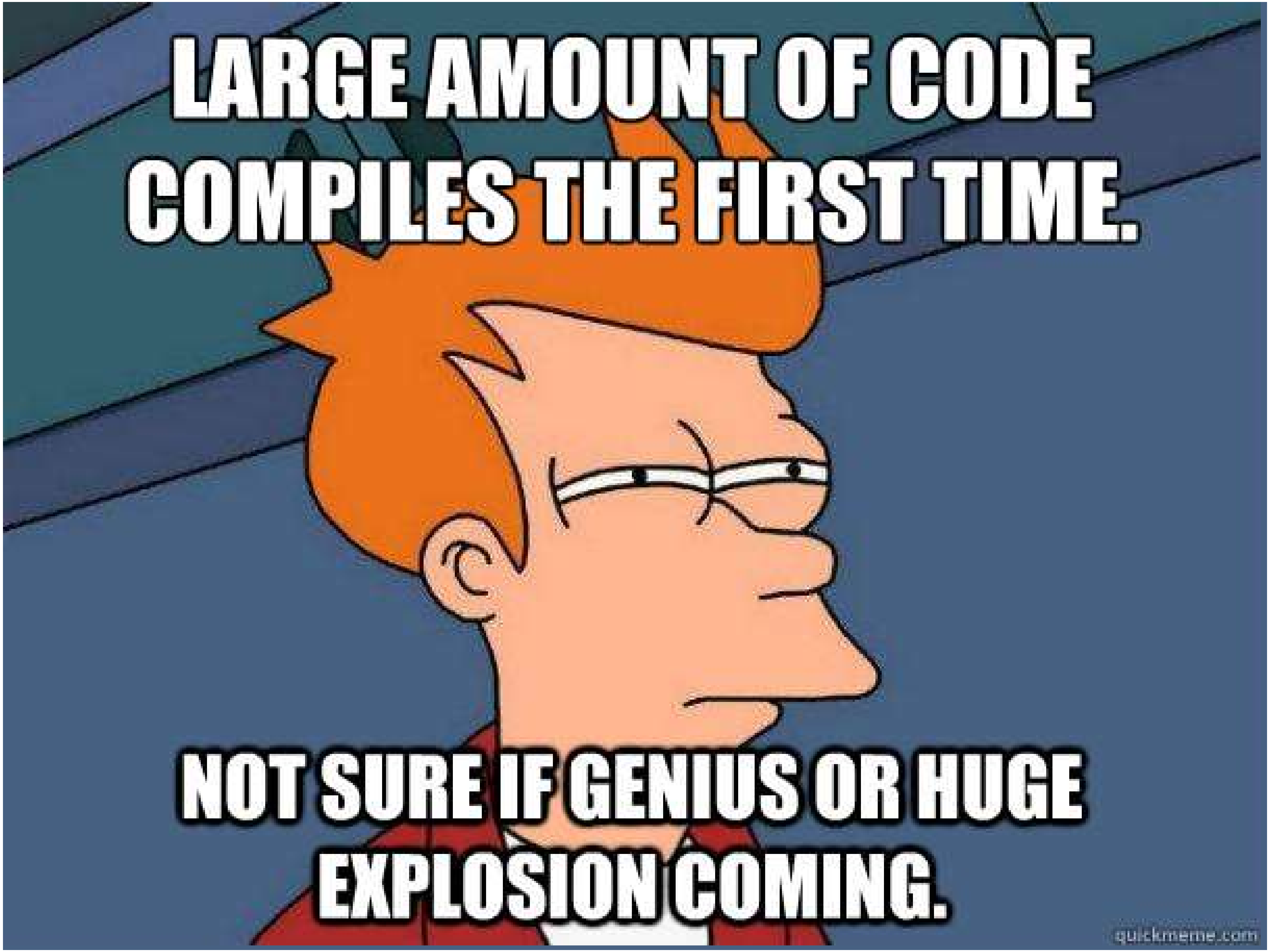
Enter valid bank code

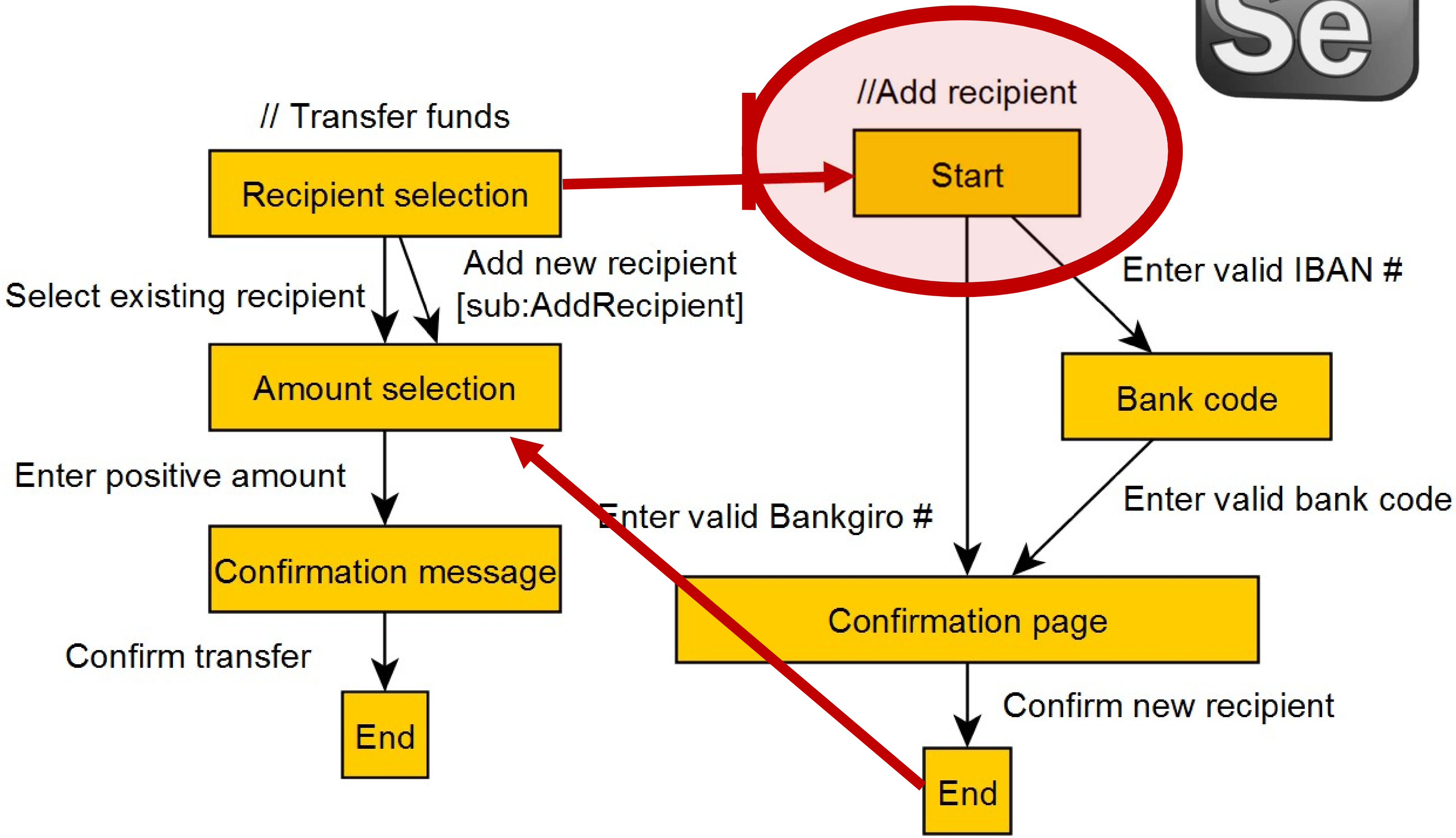
Enter valid Bankgiro #

Confirmation page

Confirm new recipient

End

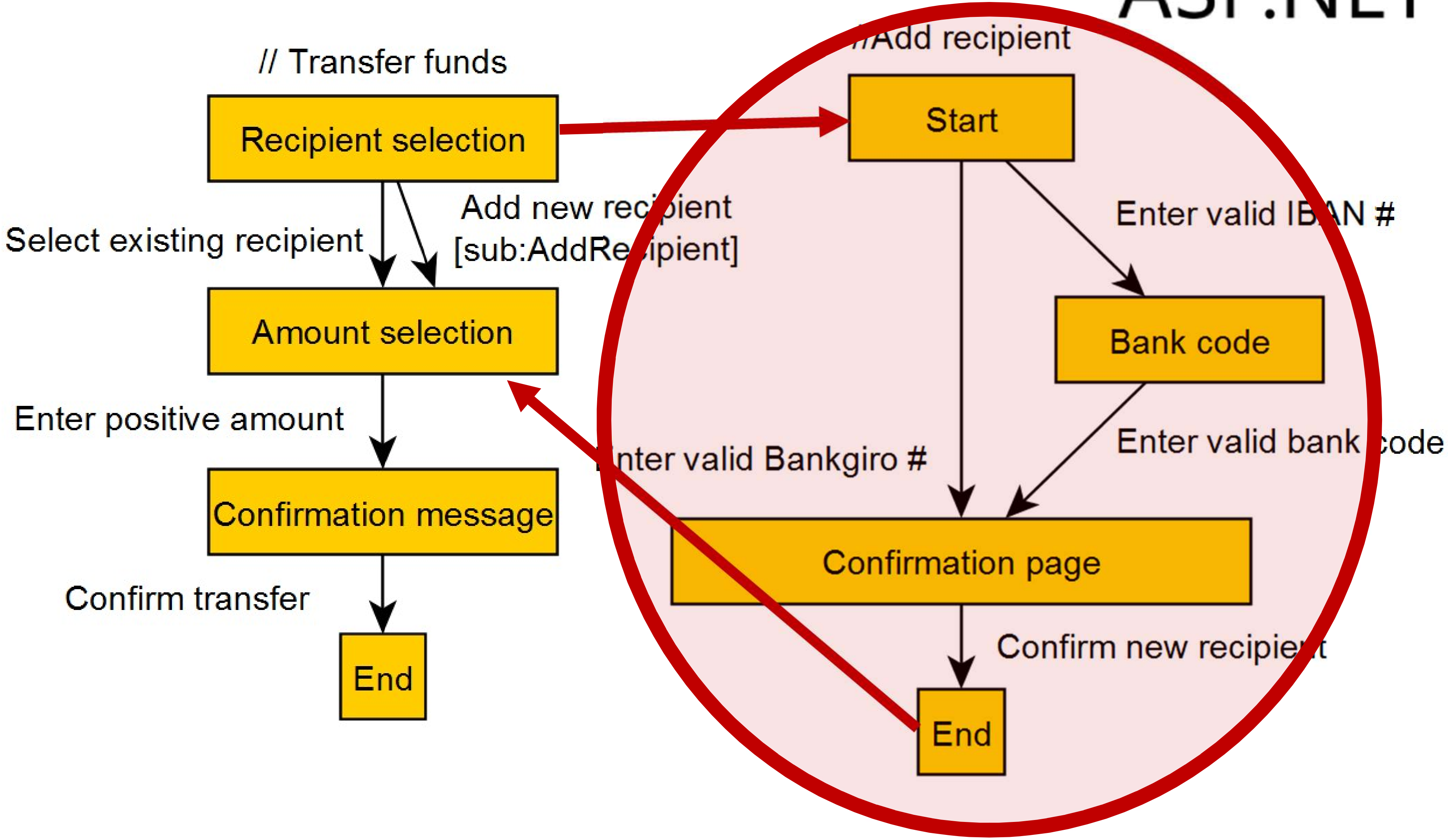

LARGE AMOUNT OF CODE COMPILES THE FIRST TIME.
NOT SURE IF GENIUS OR HUGE EXPLOSION COMING.
quickmeme.com


// Transfer funds
Recipient selection
Select existing recipient
Add new recipient
[sub:AddRecipient]
Amount selection
Enter positive amount
Confirmation message
Confirm transfer
End
//Add recipient
Start
Enter valid IBAN #
Bank code
Enter valid bank code
Enter valid Bankgiro #
Confirmation page
Confirm new recipient
End

```csharp
public abstract class PageObjectBase
{
    protected IWebDriver Driver { get; }
    protected PageObjectBase(IWebDriver driver)
    {
        Driver = driver;
    }
}
public class AddRecipientPage : PageObjectBase, IStart
{
    public AddRecipientPage(IWebDriver driver) : base(driver) {}
    public IBankCode Start_EnterValidIban()
    {
        Driver.FindElement(By.Id("txtIban"))
            .SendKeys("DE89 3704 0044 0532 0130 00");
        return new BankCodePage(Driver);
    }
    public IConfirmationPage Start_EnterValidBankGiro()
    {
        Driver.FindElement(By.Id("txtBankGiro"))
            .SendKeys("991-2346");
        return new ConfirmationPage(Driver);
    }
}
```

Microsoft

ASP.NET

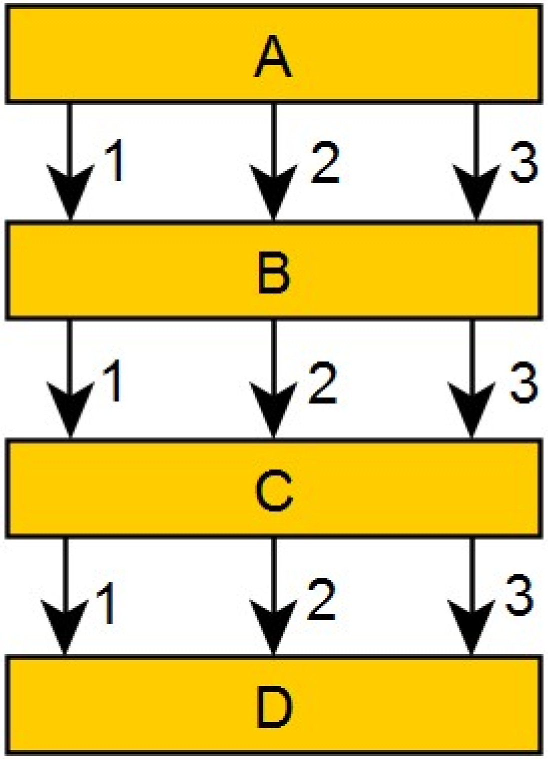


```
public abstract class MvcTestpage
{
    protected void Post(IViewModel viewmodel, string endpoint)
    { /***/ }
}
public class AddRecipient : MvcTestpage, IStart, IBankCode,
IConfirmationPage, IEnd
{
    private AddRecipientViewModel _viewModel;
    public AddRecipient()
    {
        _viewModel = new AddRecipientViewModel();
    }
    public IBankCode Start_EnterValidIban()
    {
        _viewModel.Iban = "DE89 3704 0044 0532 0130 00";
        return this;
    }
    public IEnd ConfirmationPage_ConfirmNewRecipient()
    {
        base.Post(_viewModel, "Account/Recipients");
        return this;
    }
}
```

Automated test case generation

- Test coverage criteria
  - Node coverage: 1 test
  - Edge coverage: 3 tests
  - Edge-pair coverage: 9 tests
  - All-pairs: 9 tests
  - …

Use a sufficient criteria to give confidence in the information obtained by running your tests



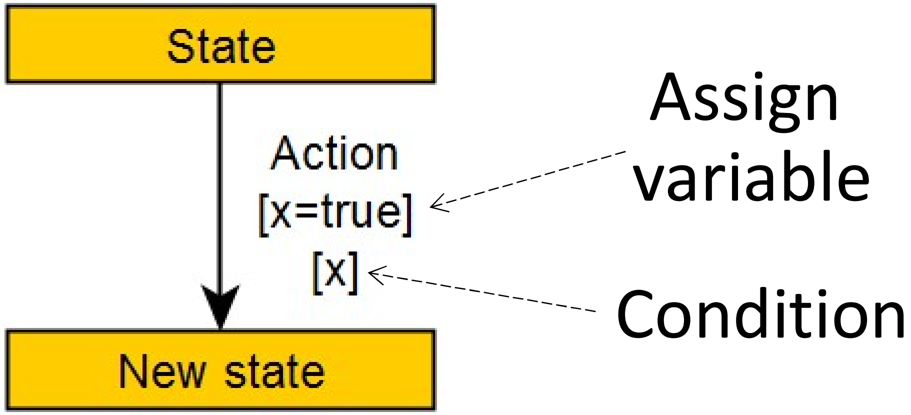
Algorithm for test case generation

- Find the path that increases coverage in the best way – Greedy!
- Repeat until no such path is found
- Warn if any edges were not reached

- Fast (~750ms on 450 edges → 50 test cases, debug)
- Good enough for us.

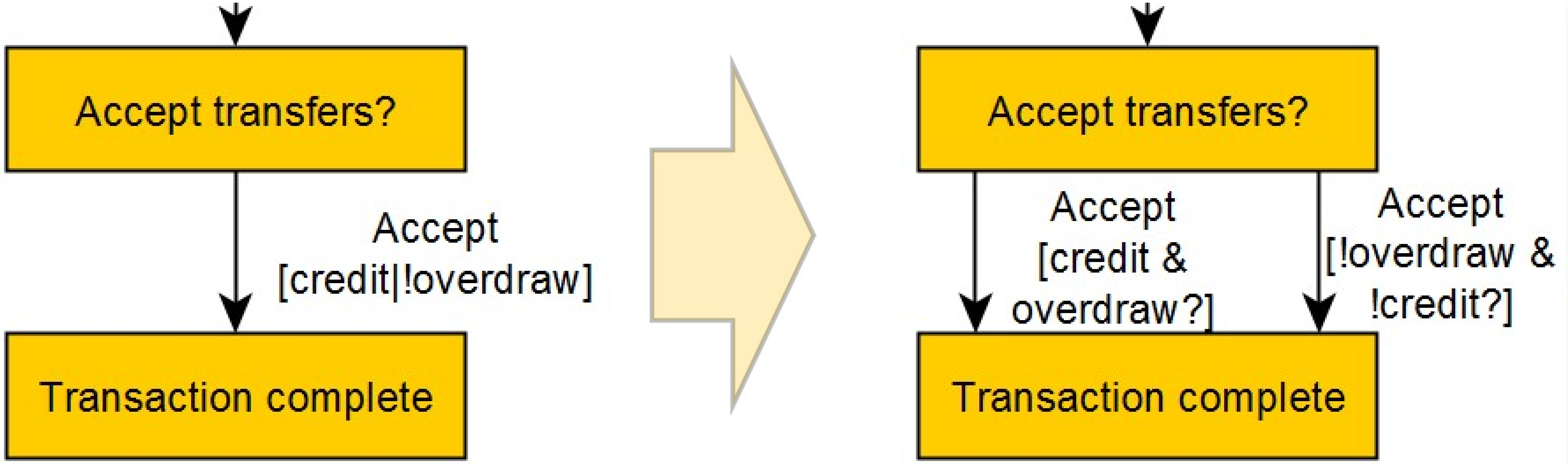
Variables and conditions

Supports basic arithmetic
(assignment, addition/subtraction and conditions)



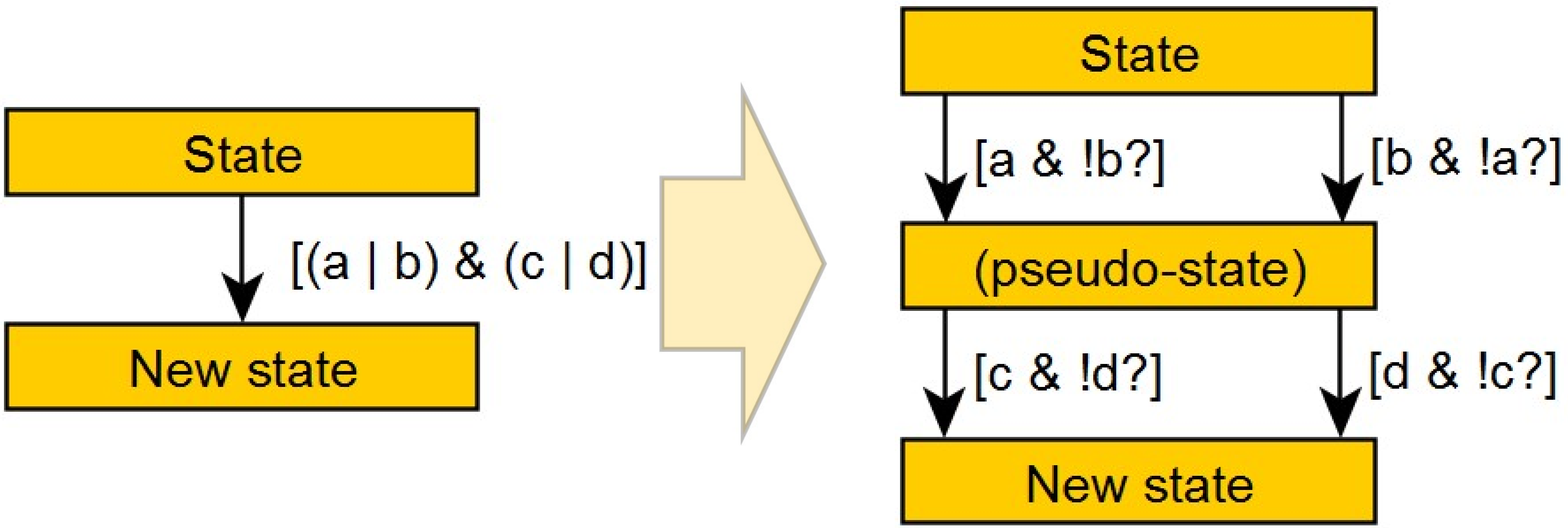
If asked for, unassigned variables are by default zero or false.

Exploding or-conditions



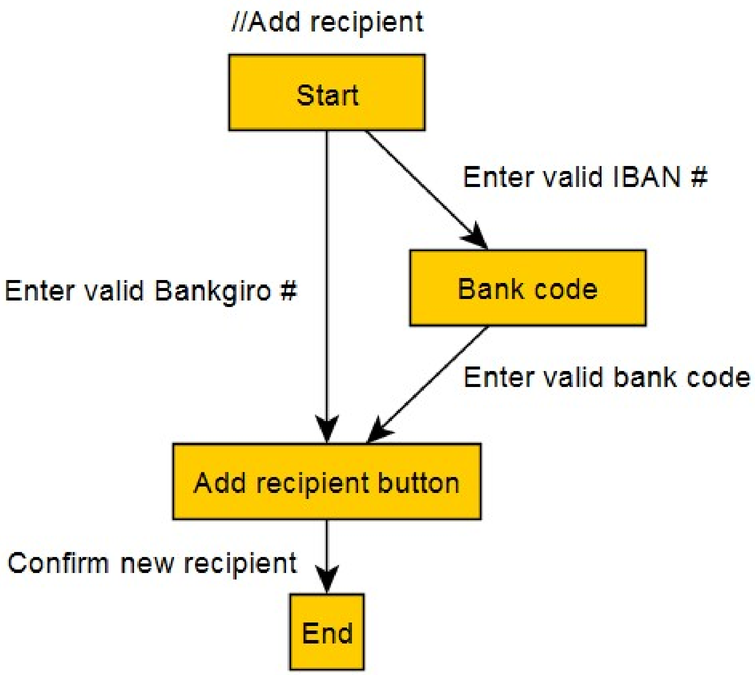
- Reversed weak conditions
  - Only checked first time edge is taken

Exploding or-conditions

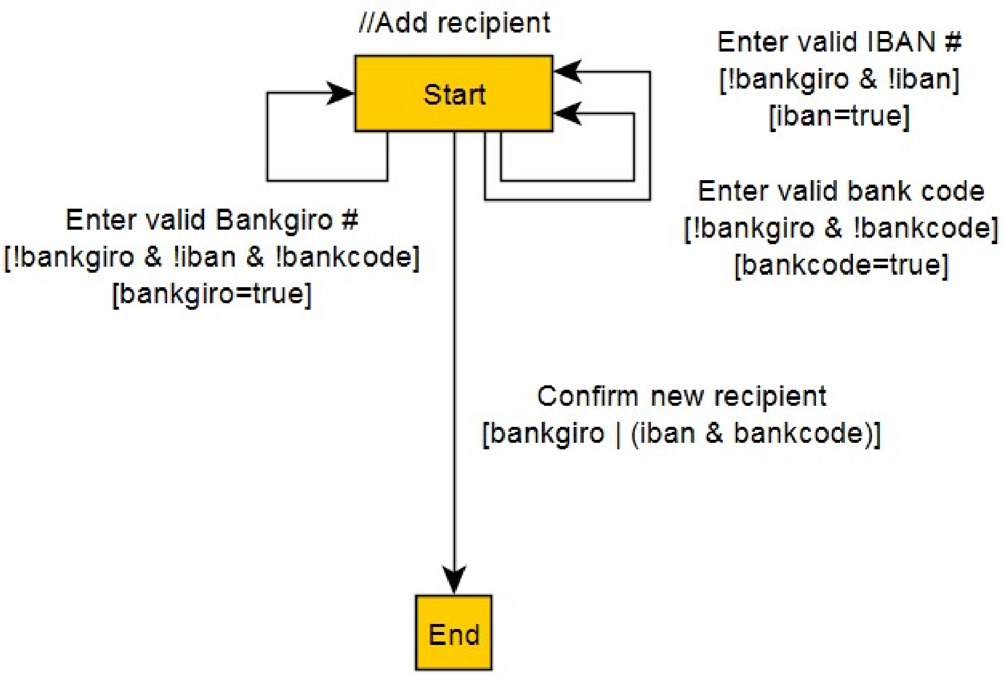


Enforced order?

- Easier to read

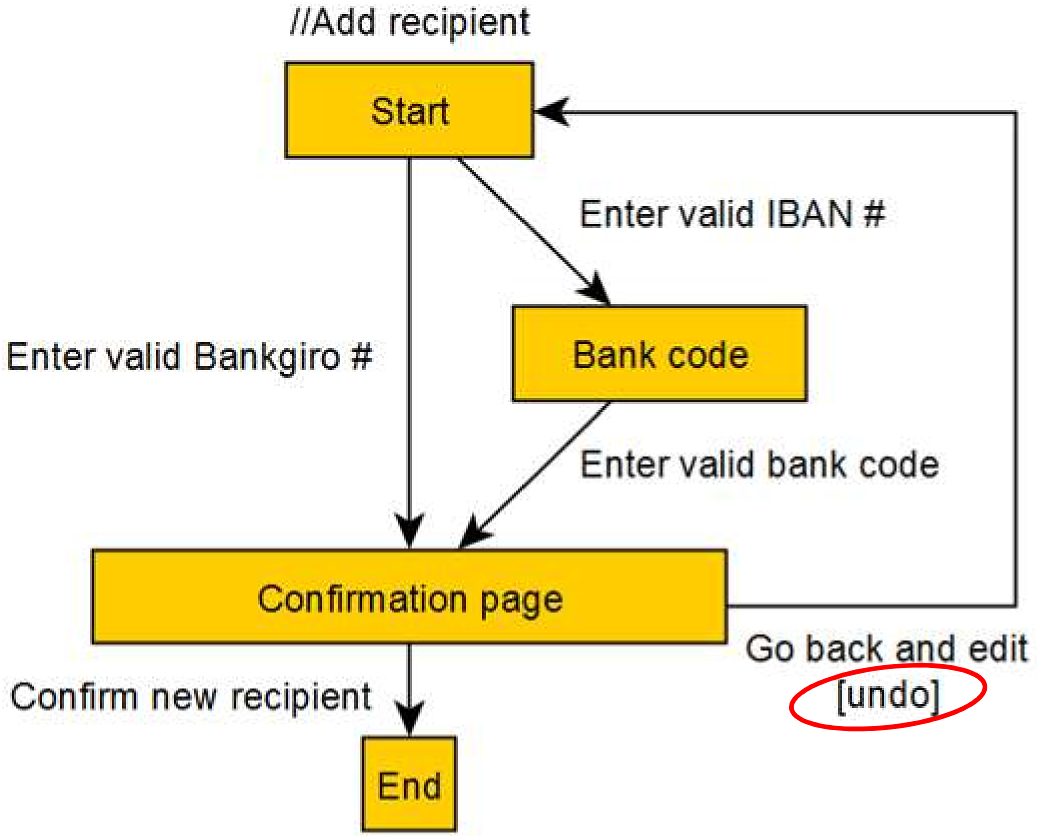


- Allowing any order of execution

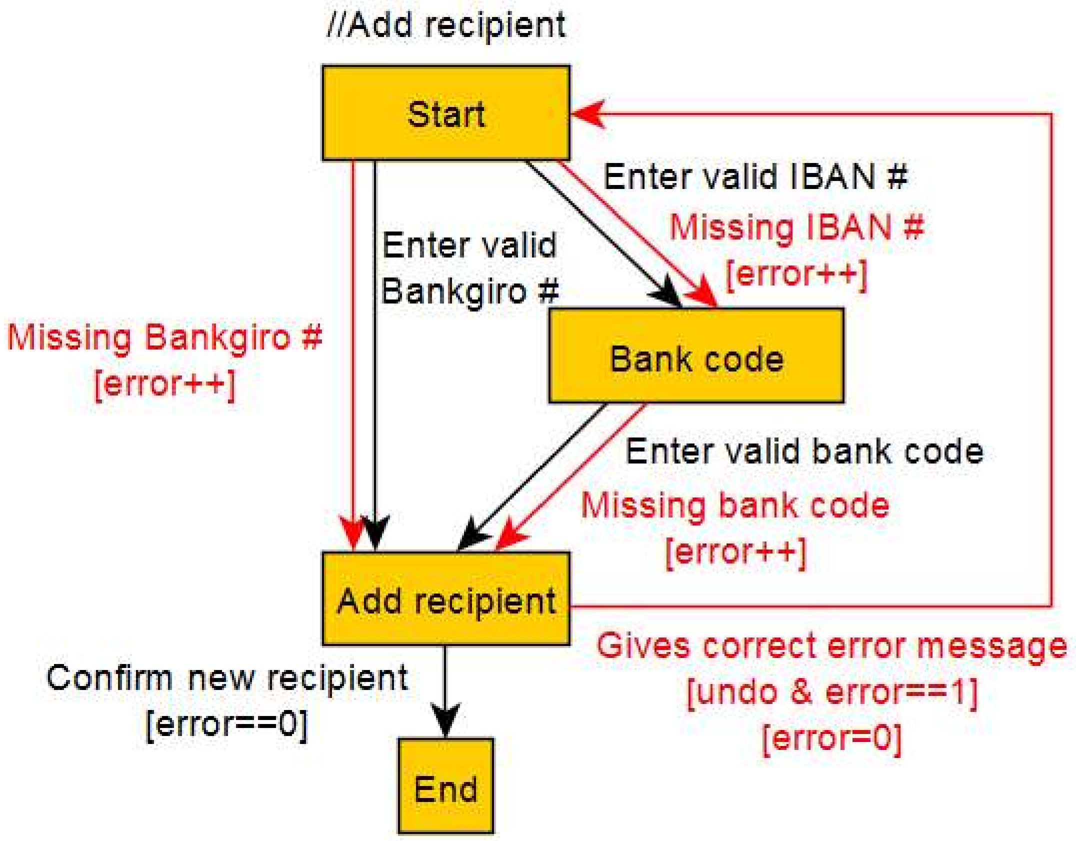


Pitfall: “Undo” Testcases

- Undo
  - Go back and edit
  - Remove something you added
  - Cancel something you are doing
- Add special condition
- Generate tests
  - Positive first
  - Then with condition activated



Negative Testcases



//Add recipient

Start

Enter valid IBAN #

Missing IBAN #
[error++]

Enter valid
Bankgiro #

Missing Bankgiro #
[error++]

Bank code

Enter valid bank code

Missing bank code
[error++]

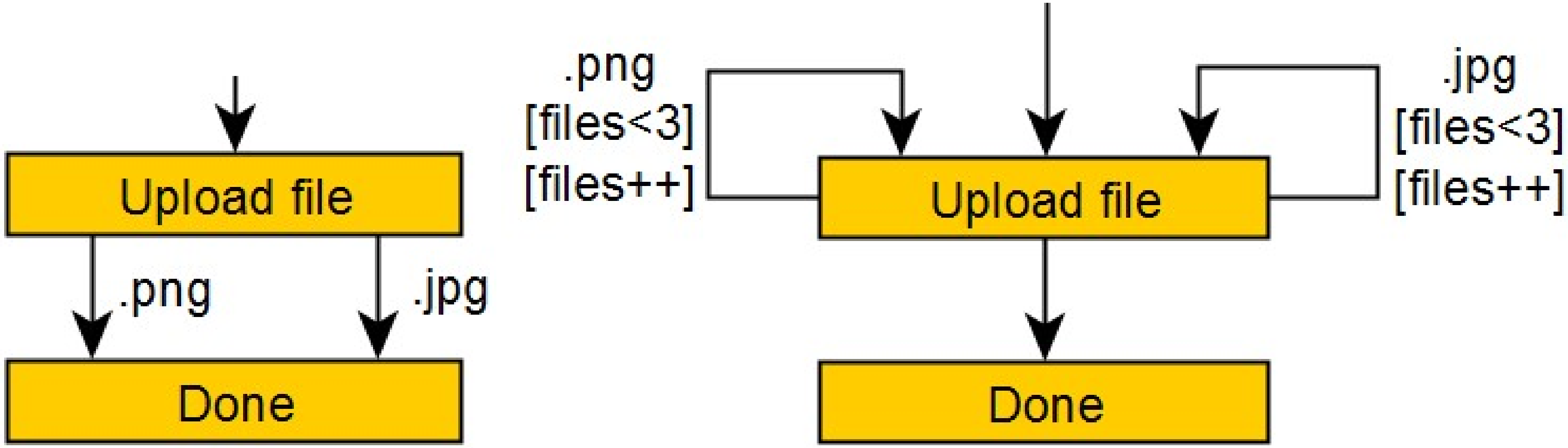
Add recipient

Gives correct error message
[undo & error==1]
[error=0]

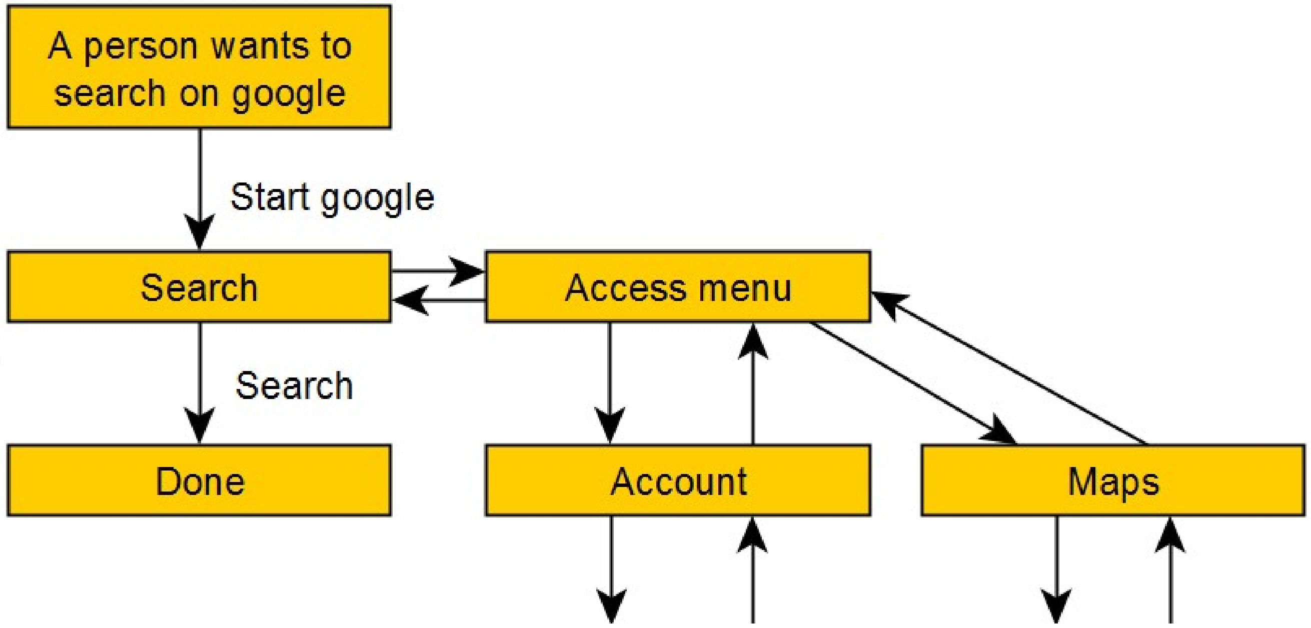
Confirm new recipient
[error==0]

End

Pitfall: Loops



Pitfall: Greedy test generation



A person wants to search on google

Start google

Search

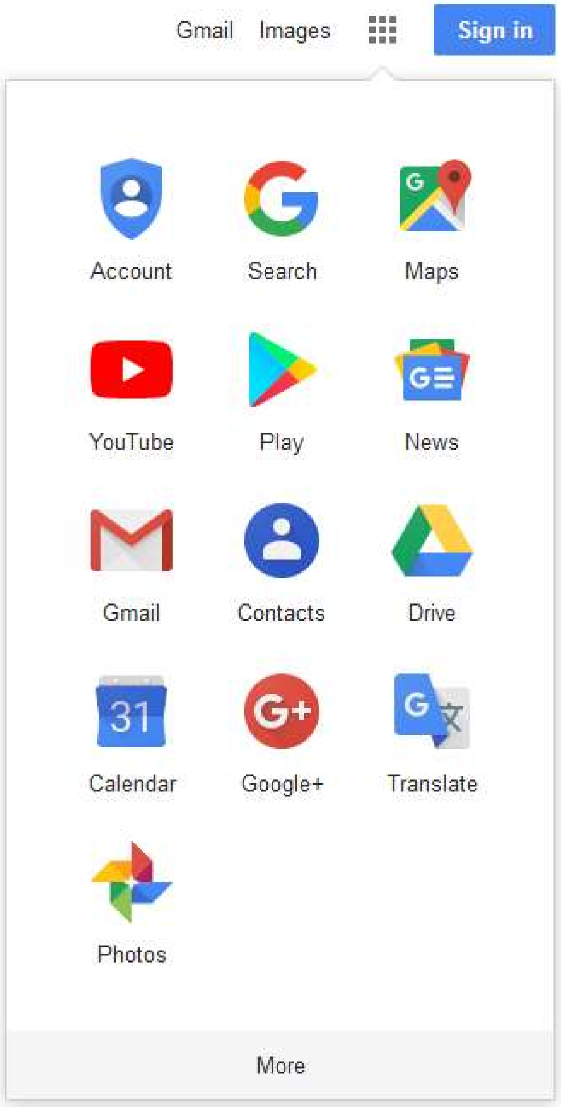
Access menu

Search

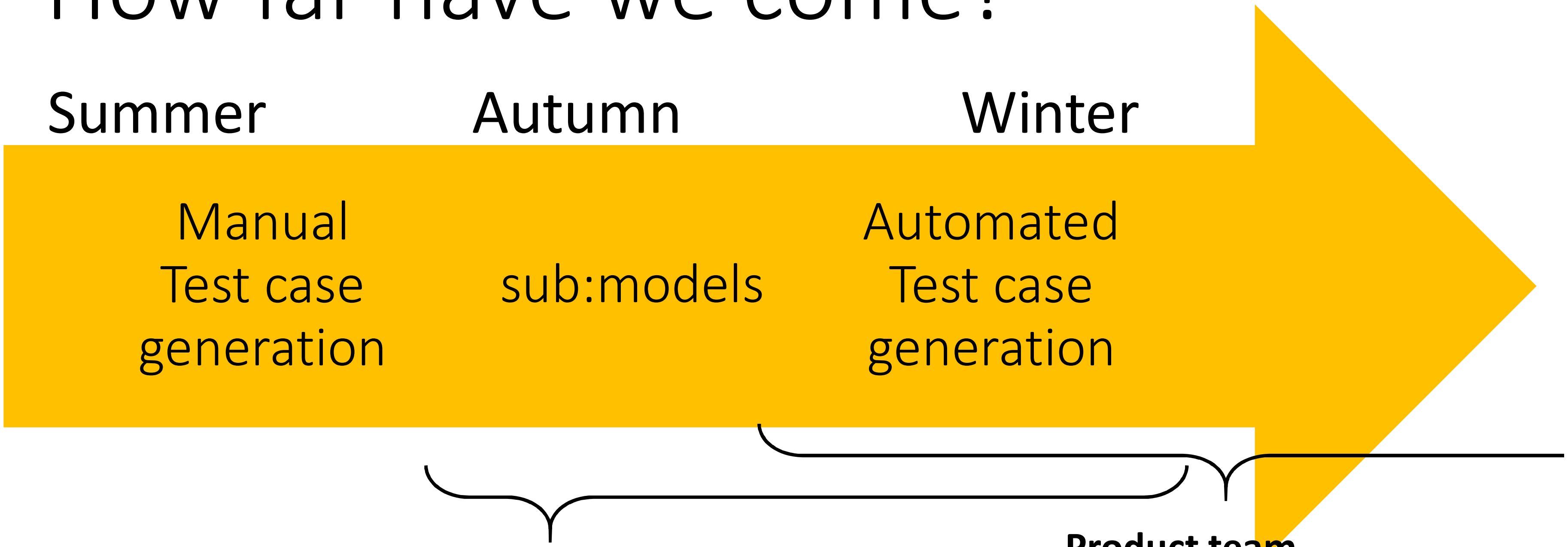
Done

Account

Maps



How far have we come?



Summer

Autumn

Winter

Manual
Test case
generation

sub:models

Automated
Test case
generation

Internship 2 students

- POC

☺

Product team

- Insurance claim reports
- Mostly web
- Multiple branches
- Models: ~40
- Nodes: ~370
- Edges: ~450
- Test cases: ~50
- Execution time: ~11 min

☺

Comparison to Graphwalker

Graphwalker

- Java implementation
- GraphML (yEd) input
- Nodes = methods (check)
- Edges = methods (do)
- Methods return void
- Random model traversal can discover unexpected bugs
- Data can be fed back to the model (online)
- Several other useful features

Our solution

- .NET implementation
- GraphML (yEd) input
- Nodes = interfaces
- Edges = methods
- Methods return next state
- Minimizes number of test cases for full edge coverage
- No feedback from execution to model
- Can generate interfaces for re-use in manual test cases

Parallel session @ 11:15

- No fixed agenda
  - More examples (code/models)?
  - Roadmap for the future?
  - Reporting?
  - You decide!

Thanks for listening!

tomasrosenqvist

johanronnlund

muamaidbengt

@muamaidbengt