

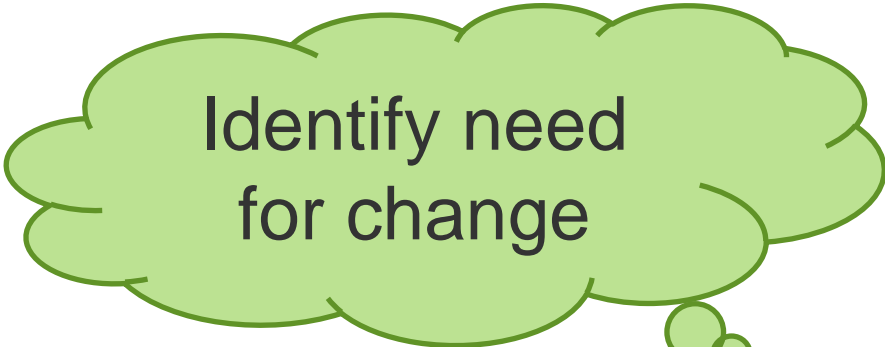
COMP33812: Software Evolution 2

Andy Carpenter

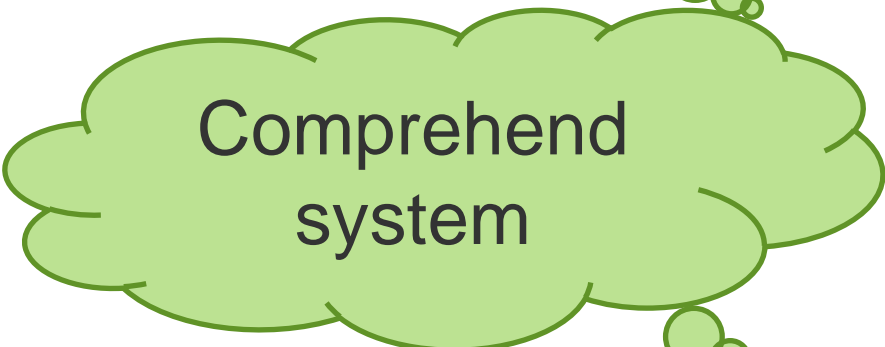
(Andy.Carpenter@manchester.ac.uk)

Elements these slides come from Sommerville, author of "Software Engineering", and are copyright Sommerville

Evolution Activities 1




Identify need
for change



Comprehend
system



Test and deploy



Get approval
for change



Make change

Evolution Activities 2

Most time?

Comprehension

40% (Sneed
1989)

90% (Standish
1984)

Comprehension

- Avoids:
 - undesirable ripple effects,
 - unintended modifications
- Breaks down to:
 - what does the system actually do?
 - where does the change need to be made?
 - how do relevant parts of the system work?
- Skill in recognising what need to understand
- Process of understanding can be modelled

Information Needs

- Driven by what need to know
- Managers: decision-support knowledge
 - cost/duration of change compared to alternatives
- Analysts: effects of change, global understanding
 - context diagrams, architecture diagrams
- Designers:
- Programmers:

Sources of Information?

- Which sources are the most reliable?

The Code Reading Process

- Formulation and testing of hypotheses about system
 - degree of certainty associated with each hypothesis
 - may have conflicted hypotheses active together
- Letovsky (1986) proposed:
 - why conjectures
 - what is the purpose of this piece of code?
 - how conjectures
 - how does this code accomplish its goal?
 - what conjectures
 - what domain role does this code element play?

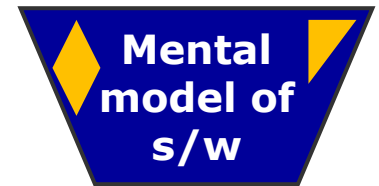
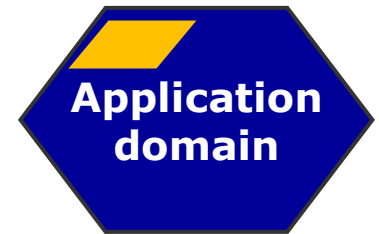
Code Reading: How is it Done?

Existing
Knowledge



How, what and why
conjectures

Acquired
Knowledge



Try it Yourself (What Questions?)

```
static void move_last_runqueue(  
    struct task_struct * p) {  
  
    struct task_struct *next = p->next_run;  
    struct task_struct *prev = p->prev_run;  
  
    next->prev_run = prev; /* remove from list */  
    prev->next_run = next;  
  
    p->next_run = &init_task; /*add back to list*/  
    init_task.prev_run = p;  
    p->prev_run = prev;  
    prev->next_run = p;  
}
```

Code Reading Strategies

- Analogues of strategies for understanding natural language texts (“reading”)
- Top-Down Strategy
 - Use context and prior assumptions to gain overall understanding from selected details
- Bottom-Up Strategy
 - Start with individual statements and build up picture incrementally
- Opportunistic Strategy
 - Switch between bottom-up and top-down as appropriate

Reading Unfamiliar Languages

- 100's of languages in use in current systems
 - C, Java, C++, F#, Ruby, COBOL, FORTRAN, VB, ...
 - Specialist languages
 - PostScript, TeX, SGML, XHTML
 - Prolog, Lisp, Scheme, Clojure
 - Groovy, REXX, ScriptEase, Perl, Python
 - RPG, SAS, MatLab, ReportWriter, awk,
 - HTML, SQL, JavaScript, VRML
- Applications often built using several languages – why?
- You will work with languages you have no training in

Code Reading: How is it Done?

Existing
Knowledge



Acquired
Knowledge



Yet Another Kind of Hypothesis

- For unfamiliar language:
 - follow the normal code reading process, plus
 - use additional class of hypothesis (about language)
- e.g.
 - “I guess that this type is an integer variable type”
 - “This group of keywords looks like an if-then-else statement”
 - “It looks as though the `&&` operator is a logical AND operator with McCarthy semantics”

Yet Another Kind of Hypothesis

- For unfamiliar language:
 - follow the normal code reading process, plus
 - use additional class of hypothesis (about language)
- e.g.
 - “I guess that this type is an integer variable type”
 - “This group of keywords looks like an if-then-else statement”
 - “It looks as though the `&&` operator is a logical AND operator with McCarthy semantics”

```
if (ptr != null
    && ptr.value > x) {
    ...
}
```

```
if (ptr != null) {
    if (ptr.value > x) {
        ...
    }
}
```

Example: Data Structures

- Data declarations
 - Name of the var/data structure should be given
 - Type may be specified
 - Implicit or inferred by some languages
 - Initial value may be specified
 - Implicit or inferred by some languages
- Data types
 - Primitive data types – examples?
 - Complex data structures – examples?

Try

- The following fragments of code are taken from the DEBES system
 - Originally designed for (the then) US Department of Agriculture
 - Coordinates soliciting and selection of bids for delivery of “food assistance”
 - e.g. school lunches, prison food, etc.

DATA DIVISION.

WORKING-STORAGE SECTION.

*

01 DATE-WORKFIELDS.

05 WS-TODAYS-DATE PIC X(06) .

05 WS-TIME PIC X(08) .

01 WORK-AREA.

05 WS-NULL-IND PIC S9(04) COMP .

05 WS-CONSTRAINT-NBR PIC S9(03) COMP-3 VALUE +0 .

05 WS-BID-COUNT PIC S9(03) COMP-3 VALUE +0 .

05 WS-TO-PROCESS-COUNT PIC S9(03) COMP-3 VALUE +0 .

05 WS-SENT-COUNT PIC S9(03) COMP-3 VALUE +0 .

05 WS-TOTAL-COUNT PIC S9(03) COMP-3 VALUE +0 .

05 WS-ALL-BIDDERS-COUNTED PIC X(01) VALUE 'N' .

```

05 WS-USER-ID.
    10 WS-USER-ENTY-CAT-CD    PIC X(01) .
    10 WS-USER-CD             PIC 9(04) .
    10 WS-USER-PLNT-CD       PIC X(02) .

```

01 WS-LITERALS.

```

05 WS-LIT-8-SPACES          PIC X(08)  VALUE SPACES.
05 WS-LIT-2-SPACES          PIC X(02)  VALUE SPACES.
05 WS-LIT-MDBC9003          PIC X(8)   VALUE 'MDBC9003'.

```

01 H-HTML-WEBPAGE.

```

02 FILLER          PIC X(6)    VALUE '<HTML>'.
02 FILLER          PIC X(1)    VALUE X'25'.
02 FILLER          PIC X(57)   VALUE
    '<BODY onLoad = "top.frames[2].' -
    'findform(this, ''dontsave'', '

```

```

02 H-HTML-RETURN-CODE      PIC ZZZ9.

```

```

02 FILLER          PIC X(3)    VALUE ')">'.

```

```

02 FILLER          PIC X(18)   VALUE '<TABLE width=100%>' .

```

Example: Statements and Control

- Assignments
- Expressions
- Calls to other code

- Sequence
- Alternation
- Iteration

* MAIN PROGRAM ROUTINE

* PERFORM HOUSEKEEPING, PERFORM WRAP UP PROCESSES

0000-MAINLINE-MDBC740D.

MOVE L-RETURN-CODE TO H-HTML-RETURN-CODE.

PERFORM 4100-BUILD-INVIT-DESCR

THRU 4100-BUILD-INVIT-DESCR-EXIT.

IF H-INVT-STAT = WS-LIT-OP OR H-INVT-STAT = WS-LIT-SP

PERFORM 0100-MAIN-PROCESS

THRU 0100-MAIN-PROCESS-EXIT

ELSE

MOVE ZEROES TO H-RETURN-CD2

CALL WS-LIT-MDBC9003 USING H-STATUS-ERR

BY CONTENT LENGTH OF H-STATUS-ERR

END-IF.

0000-MAINLINE-MDBC740D-EXIT.

GOBACK.

Program 3

```
<h1>Add/Edit Competition</h1>
<?php
    echo Form::open(
        Route::get('ManageCompetitions')->uri(array(
            'action' => 'addEditCompetition',
            'param' => 'submit'
        )) . URL::query($query_parameters, FALSE),
        array(
            'id' => 'addEditCompetition',
            'class' => 'input_form' ...,
            'enctype' => 'multipart/form-data'
        )) . PHP_EOL;
?>
<ul>
```

Program 3

```
<script type="text/javascript"
    src="../../../angularjs/1.3.12/angular.min.js" />
<script type="text/javascript">
    var SPS = angular.module('SPS', []);
    SPS.controller('controller',
        function ($scope, $http) {
            $http.get('...').success(function(data) {
                $scope.event_options = data;
            });
        });
</script>
<section ng-controller="controller">
    <select name="event_id" ng-model="event_selected">
        <option ng-repeat="event in event_options"
            value="{{event.id}}">{{event.label}}</option>
    </select>
```

Program 4

```
[comment encoding = UTF-8 /]
[module pageController(
    'http://www.cs.man.ac.uk/mdsd/2010/GenJsf', ...)]

[template public controller(genPage : GenPage)]
[file(genPage.controllerFilename(), false)]
<?php defined('SYSPATH') ...
class [genPage.controllerClassName()] extends
[genPage.baseControllerClassName()]
{
    public function before()
    {
[if (genPage.isAuthenticated())]
        [genPage.loggedInCheck()]
[/if]
        parent::before();
    }
}
```

Program 4

```
[template public unitMethods (genUnit :  
  GenDynamicUnit)  
  ? (genUnit.oclIsKindOf (GenIndexUnit))  
post(trim())]  
[let genIndexUnit : GenIndexUnit  
  = genUnit.oclAsType (GenIndexUnit)]  
[for (genAction : GenDeleteAction  
  | genUnit.oclAsType (GenIndexUnit).deleteActions())]  
[genAction.generateActionMethod('$this->  
  .concat(genIndexUnit.instanceName())  
  .concat('_error')))/]  
[/for]  
[/let]  
[/template]
```


Program 5

```
module exam;
create jsf : JSF from website : Website, orm : ORM;

helper def: findOrmEntity(name : String) : ORM!Entity
    = ORM!Entity.allInstancesFrom('orm')->any(e |
e.name = name);

abstract rule NamedElement {
from website : Website!NamedElement
to jsf : JSF!NamedElement (
    name <- website.name
)
}
```

Program 5

```
-- @extends NamedElement
rule Service {
  from website : Website!Service
  to jsf : JSF!Service (
    modelName <-
      if not website.modelName.oclIsUndefined() then
        website.modelName
      else
        website.encapsulates->first().name
      endif,
  ...
}
```

Program 5

...

```
entities <-  
  website.encapsulates  
    ->collect(e | thisModule.findOrmEntity(e.name)) ,  
  displayLabels <- website.displayLabels,  
  selections <- website.selections,  
  includedFeatures <- website.includedFeatures  
)
```

Program 6

```
class Entity extends Classifier,UnitSource {  
  property features#parentEntity  
    : Feature[*] { ordered composes };  
  property associationEnds#targetEntity  
    : Association[*] { ordered };  
  property servedBy#encapsulates  
    : Service[*] { ordered };  
  property displayFeatures : Feature[*] { ordered };  
  attribute displayFormat : String[?];  
  invariant featureNameUniqueWithinEntity:  
    features->isUnique(name);  
  invariant displayOnlyLocalFeatures:  
    features->union(associationEnds)  
      ->includesAll(displayFeatures);  
}
```

Program 6

Invariant includedFeaturesMustBeFromEncapsulated:

```
let features : Sequence (Feature)
  = encapsulates->collect(e | e.features)
  ->union(
    encapsulates->collect(e | e.associationEnds))
in features
->select(f | f.oclIsKindOf(IncludedFeature))
->forall(f | features
->includes(
  if f.oclIsTypeOf(ServiceEntityElement) then
    f.oclAsType(ServiceEntityElement).feature
  else
    f.oclAsType(ServiceEntityAssociation).feature
endif)) ;
```

Program 6

```
attribute name : String[?] { derived readonly volatile }  
{  
  derivation: if feature.ocllsUndefined() then  
    "  
    else if self.alias.ocllsUndefined() then  
      feature.name  
    else if self.alias <> " then  
      self.alias  
    else  
      feature.name  
    endif endif endif;  
}
```

Overview of COMP33812

- System Evolution
 - making the changes above the code level
- A/B Testing
 - evaluation of alternative changes
- Software Quality
 - How good is our code? How do we know?
- Software reuse
 - exploiting others quality
- Future
 - ways to simplify system comprehension

Assessment

- 70% Examination, 30% coursework
- Coursework:
 - Three exercises
 - Details on Blackboard
 - Deadlines 17.00 Friday week 3, 6 and 9
 - First considering usefulness of old studies
 - Second is application of testing
 - Third ...