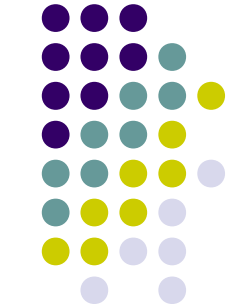
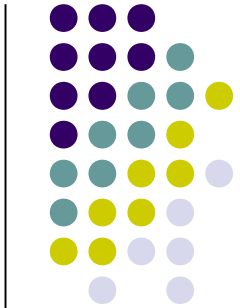
Code Complexity

Code Measures



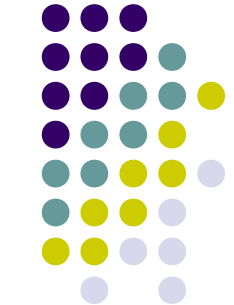
- Most common measures
  - The artifact is almost always available
  - Automatic extraction (surveys are expensive)
  - Less bias if there is no incentive to cheat
- For example
  - Size
  - Complexity
  - Quality
  - Understandability
  - Maintainability

Size and Churn



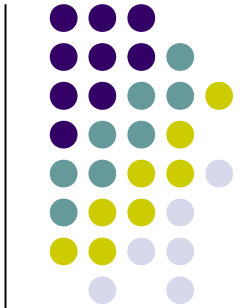
- Thousands of Lines Of Code (KLOC)
  - Usually excluding comment (NCSL)
- Churn
  - Number of lines of code changed between versions
  - Useful for maintenance
- Dependent on programming language
- Simplistic
- Correlates strongly with other measures
  - Effort and complexity

Complexity

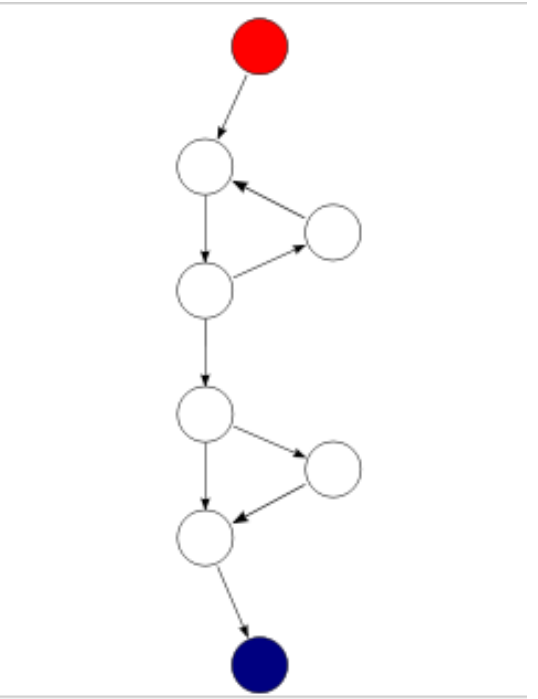


- Complexity of maintenance task
  - “difficulty of maintaining, changing, and understanding programs”, Zuse
  - More complex a system is the more difficult it is to understand and change
- No single measure of system complexity
  - Program structure, semantic content, data flow, and algorithmic complexity
  - 100's of measures in the literature

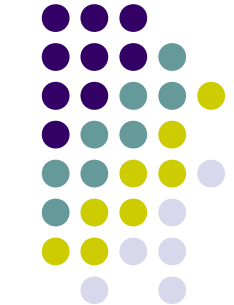
McCabe's Cyclomatic Complexity



- Number of independent paths through a program
  - v(F) = e – n + 2P
    - n is total nodes, e is total number of edges,
    - v(F) is the cyclomatic number
    - P is number of connected components
- No account of complexity of conditionals or nested loops



Halstead's Measures

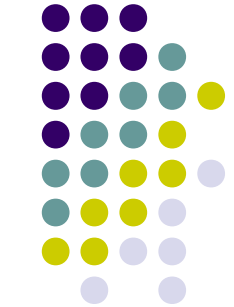


- Counts of operators
  - Change value, e.g., +/-*, while, etc
- and operands
  - Variables or const
- For example,
  - Total number of unique operands and operators
  - Difficulty, volume, vocabulary, effort
- Problems
  - Poorly defined
  - Poorly controlled studies on unrealistic software projects

Correlation

- The strength of the relationship between measures
  - e.g., demand for a product and its price
  - Departure from independence
- Generally accepted levels (between -1 and 1)
  - Strong .7 - 1.0
  - Moderate .5 - .7
  - Weak .3 - .5
  - Very weak .1 - .3
- What does a negative correlation mean?

But look at the correlations! [Herraiz et al]

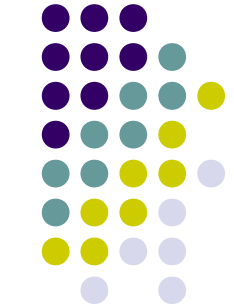


	SLOC	LOC	FUNC
SLOC	1		
LOC	9844	1	
FUNC	8216	8187	1
BLKL	9186	9364	8203
CMTL	6742	7676	5477
CMTN	7612	7991	6018
CYCLO	9325	9164	7856
RETUR	7496	7469	7498
HLENG	9817	9655	7942
HVOLU	9815	9652	7921
HLEVE	9033	8851	7005
HMD	9715	8882	7737

Parsimony is the scientific idea that the simplest explanation of a phenomenon is the best one.

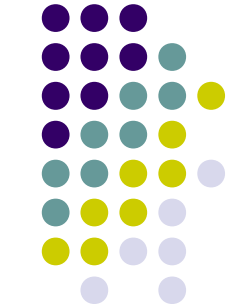
Size	Source Lines of Code (SLOC), Lines of Code (LOC), Number of C functions (FUNC), Number of blank lines (BLKL), Number of comment lines (CMTL), Number of comments (CMTN)
Complexity	McCabe's cyclomatic complexity (CYCLO), Number of function returns (RETUR), Halstead's length (HLENG), Halstead's volume (HVOLU) Halstead's level (HLEVE), Halstead's mental discriminations (HMD)

Gaming the system



- If you are paid for every line of code, and you are greedy, what would you do?
- OS/2
  - “Microsoft programmers also became frustrated with IBM's bureaucracy and its use of lines of code to measure programmer productivity”
- Normally not an issue, so use the simplest measure!

Complexity of a change

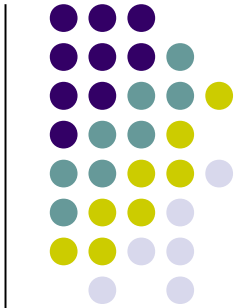


- System complexity
  - vs
- Change complexity

Simple complexity measures

- Complexity measures were designed to work on the system as a whole
  - Can be reimplemented, but …
- Why not start simple
  - Churn
  - Number of diffs and files changed
  - Diff hunks
    - Line location within file
  - Directory distance between files
    - Proxy for architecture
  - Level of indentation

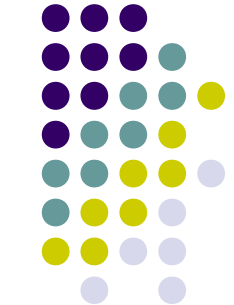
```
@@ -101,10 +101,10 @@ static int iommu_queue_command(struct amd_iommu *iommu, struct iommu_cmd *cmd)
  */
 static int iommu_completion_wait(struct amd_iommu *iommu)
 {
-       int ret, ready = 0;
+       int ret = 0, ready = 0;
        unsigned status = 0;
        struct iommu_cmd cmd;
-       unsigned long i = 0;
+       unsigned long flags, i = 0;

        memset(&cmd, 0, sizeof(cmd));
        cmd.data[0] = CMD_COMPL_WAIT_INT_MASK;
@@ -112,10 +112,12 @@ static int iommu_completion_wait(struct amd_iommu *iommu)

        iommu->need_sync = 0;

-       ret = iommu_queue_command(iommu, &cmd);
+       spin_lock_irqsave(&iommu->lock, flags);
+
+       ret = __iommu_queue_command(iommu, &cmd);

        if (ret)
-               return ret;
+               goto out;
```



Which change is more complicated? Above or Below?

```
@@ -235,7 +235,7 @@ static void tick_do_broadcast_on_off(void *why)
        case CLOCK_EVT_NOTIFY_BROADCAST_FORCE:
                if (!cpu_isset(cpu, tick_broadcast_mask)) {
                        cpu_set(cpu, tick_broadcast_mask);
-                       if (td->mode == TICKDEV_MODE_PERIODIC)
+                       if (bc->mode == TICKDEV_MODE_PERIODIC)
                                clockevents_shutdown(dev);
                }
                if (*reason == CLOCK_EVT_NOTIFY_BROADCAST_FORCE)
```

Level of Indentation

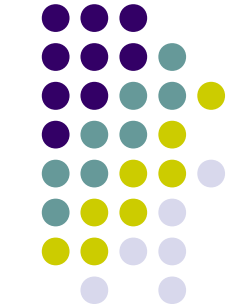


- The deeper the indentation, the more complex the code
- High correlation with other complexity measures!

```
1 > void square(int * arr, int n) {
2 >     int i = 0;
3 >     for ( i = 0 ; i < n ; i++ ) {
4 >         arr[i] *= arr[i];
5 >     }
6 > }
```

Metric	Raw	Logical
LOC	6	6
AVG	3.33	0.833
MED	4	1
STD	2.75	0.687
VAR	9.07	0.567
SUM	20	5
MCC	2	2
HVOL	142	142
HDIFF	15	15
HEFFORT	2127	2127

Paper: Change vs System Complexity



- Take a system
- For each change
  - Calculate standard complexity measures (SC)
  - Calculate change complexity measures (CC)
- Compare
  - SC_N = sum_i_N(CC)
  - CC_i = SC_i – SC_i-1
- Related work: Herraiz2011QC2