Teaching Operating Systems Using Code Review

Christoffer Dall and Jason Nieh
Columbia University

Learning OS is hard

- Operating Systems are large and complicated

- Hands-on experience is crucial

- Modification to large and unfamiliar code bases

Teaching OS is hard

- Programming assignments must be reviewed

- Code is dense and spread out

- Disproportionate amount of time spent evaluating compared to teaching

GradeBoard

- Code review system designed for the classroom

- Web-based, intuitive, easy-to-use

- Build on **Git** and **ReviewBoard**

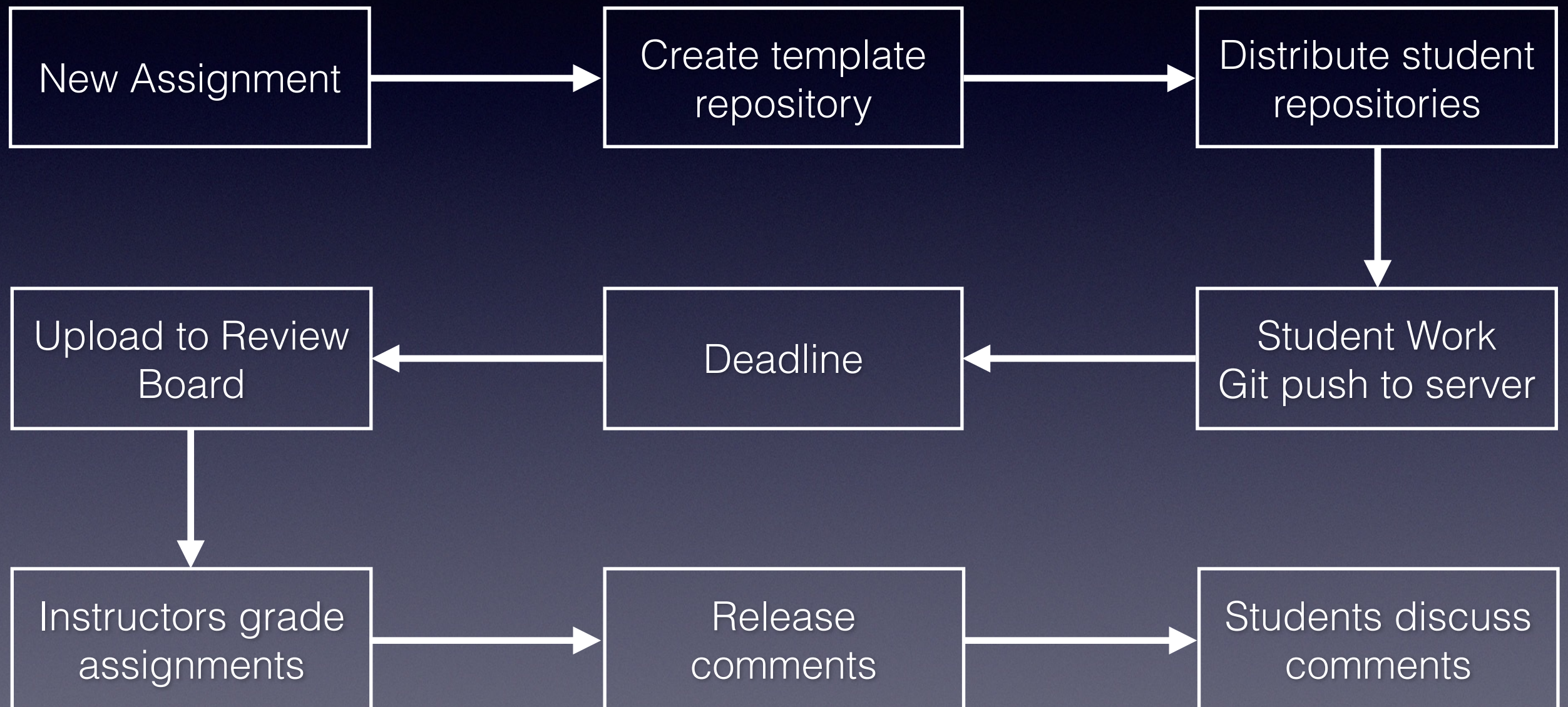- Supports entire workflow of teaching OS

Outline

- Usage Model

- System Architecture
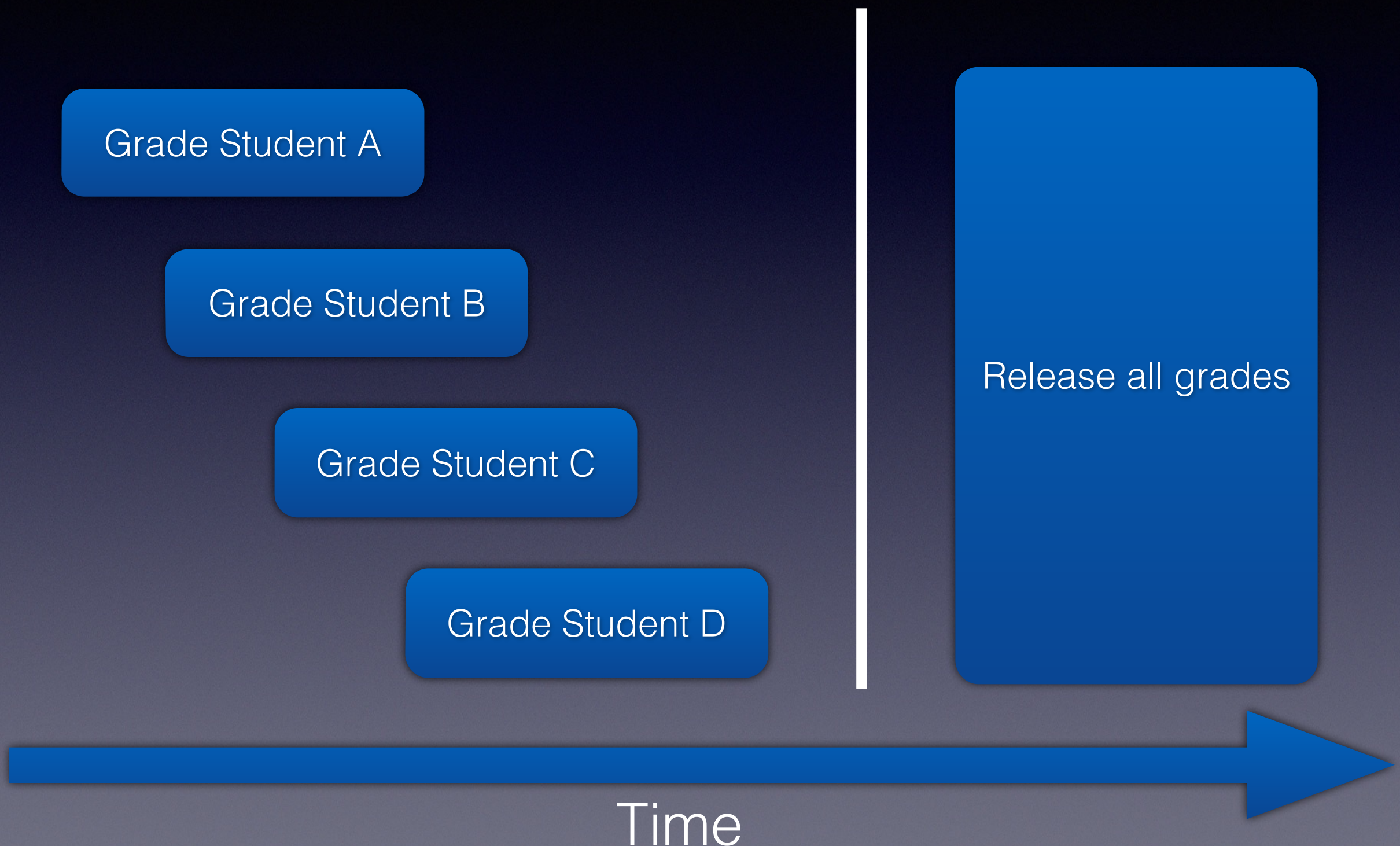
- Evaluation

- Conclusions

Usage Model

- Instructors and students

- Instructors **create**, **distribute**, and **grade** programming projects

- Students **download**, **backup**, **collaborate**, and **submit** programming projects

Workflow

```
New Assignment  →  Create template      →  Distribute student
                   repository              repositories
                                                  ↓
Upload to Review  ←  Deadline           ←  Student Work
Board                                       Git push to server
   ↓
Instructors grade  →  Release            →  Students discuss
assignments           comments              comments
```

The Web Interface

The Web Interface

Homework 5 group submissi... ✕

← → ⟳ 🌐 reviews.os1.cs.columbia.edu/r/690/diff/#index_header ☆ 🔵

M Mail   M Virtual Open System   7 Christofferdall.dk – C   🌐 NetBank   ⊞ mach_bsd_syscalls –   💹 PeterMaydell/KVM/H   ⟳ LXR linux/   📁 Resarch Libraries   »   📁 Other Bookmark

```
387 bebc6000-bebe7000 rwxp 00000000 00:00 0 ...............X......XXX11      [stack]
388
```

linux-2.6.35-cm/fs/proc/ta

Revision 271afc48b9a5d58dd28

```
...

206 static void show_map_vma(                              ile *m, struct vm_area_struct *vma
207 {
208         struct mm_struct                                m_mm;
209         struct file *file                               ile;
210         int flags = vma->
211         unsigned long ino
212         unsigned long lon
213         unsigned long sta

214         dev_t dev = 0;

215         int len;
216
217         if (file) {
218                 struct in                               vma->vm_file->f_path.dentry-

219                 dev = ino                               dev;
220                 ino = ino
221                 pgoff = (                               >vm_pgoff) << PAGE_SHIFT;
222         }

...

232                 v                                       AD ? 'r' : '-',
233                 f                                       ITE ? 'w' : '-',
234                 f                                       EC ? 'x' : '-',
235                 f                                       YSHARE ? 's' : 'p',
236                 f
237                 P
238                 M                                       INOR(dev), ino, &len);
239
```

Other reviews

Christoffer Dall [View] [Reply]

you need to handle page_mapcount == (think about the ZERO_PAGE)

Points deducted: 3

Team14 [View] [Reply]

Do you mean we should have printed 1 page_mapcount == 0. We felt it is wrong as user cannot find whether it is ZERO_PAGE or one page table entry pointing to the physical page. Could you clarify how we need to handle page_mapcount == 0 in different way without removing the meaning of each value mapcount can take ?

Christoffer Dall [View] [Reply]

just check for mapcount == 0 and pri you know that at least the process printing has a reference, so 0 is not the case. The ZERO_PAGE is an optimization, and the mapping will be changed to a dedicated page once you write to that page, so the analogy is having a single dedicated page referenced.

The homework does not allow you to print 0, and my guess is that you did not even think about the fact that page_mapcount can be 0.

You also have no comment explaining this.

Team14 [View] [Reply]

We discussed about ZERO_PAGE due to with TA(Abhijeet during our demo). Forgot to include comment regarding that in code. Hence in the test case 2, to get 1's in all pages we write to all pages we have mmaped. However irrespective of this, printing 1 in case mapcount == 0 is wrong. The process that is printing has only read reference. Multiple process/threads could

Your comment

Points to subtract [0]   Subtract   Reset

[Save] [Cancel]

```
241
242
243                 seq_printf(m, "%s", " ");
244         for (addr = start; addr < vma->vm_end; addr += PAGE_SIZE) {
245                 page = follow_page(vma, addr, FOLL_TOUCH);
246                 if (IS_ERR_OR_NULL(page))
247                         seq_printf(m, "%s", ".");
```
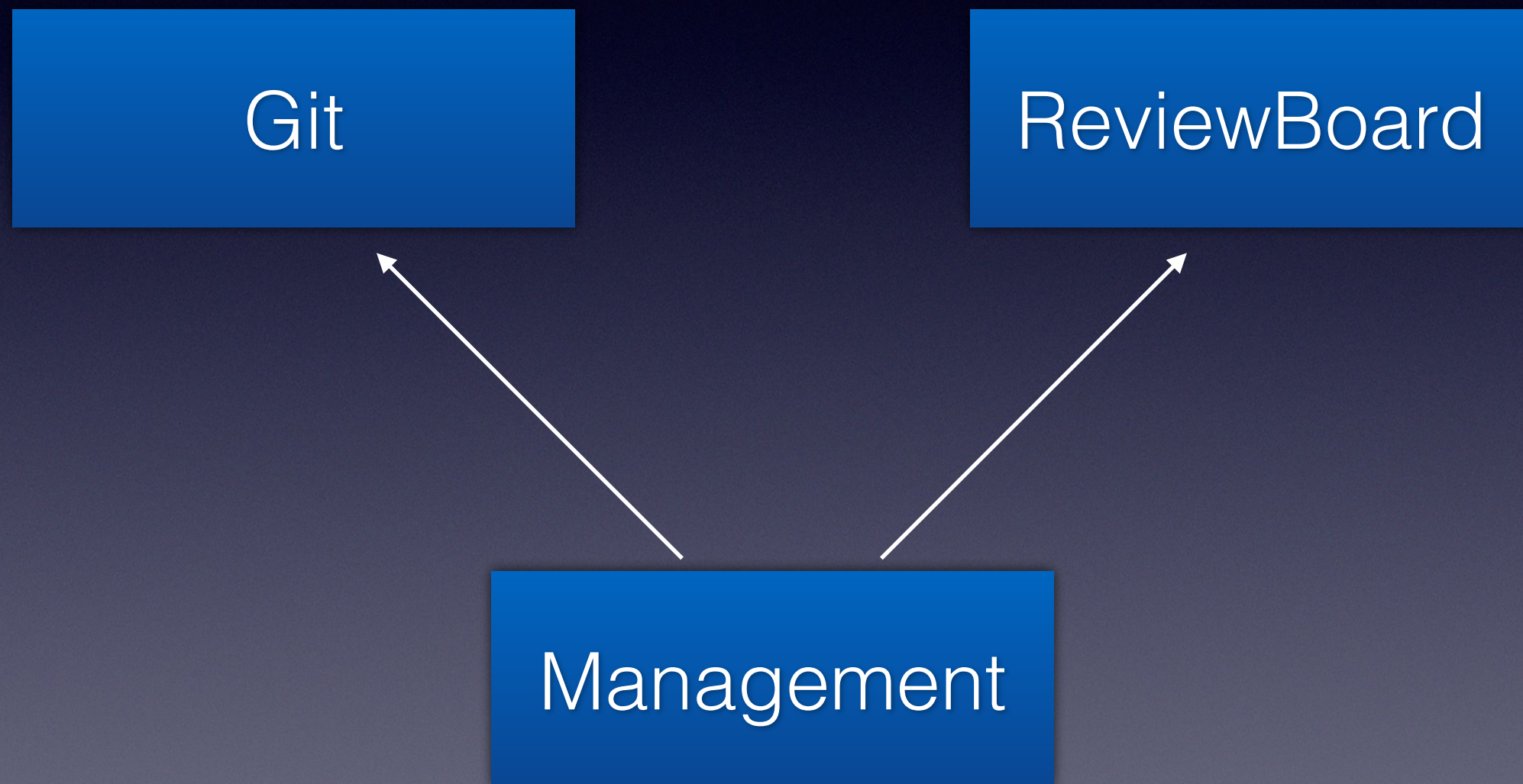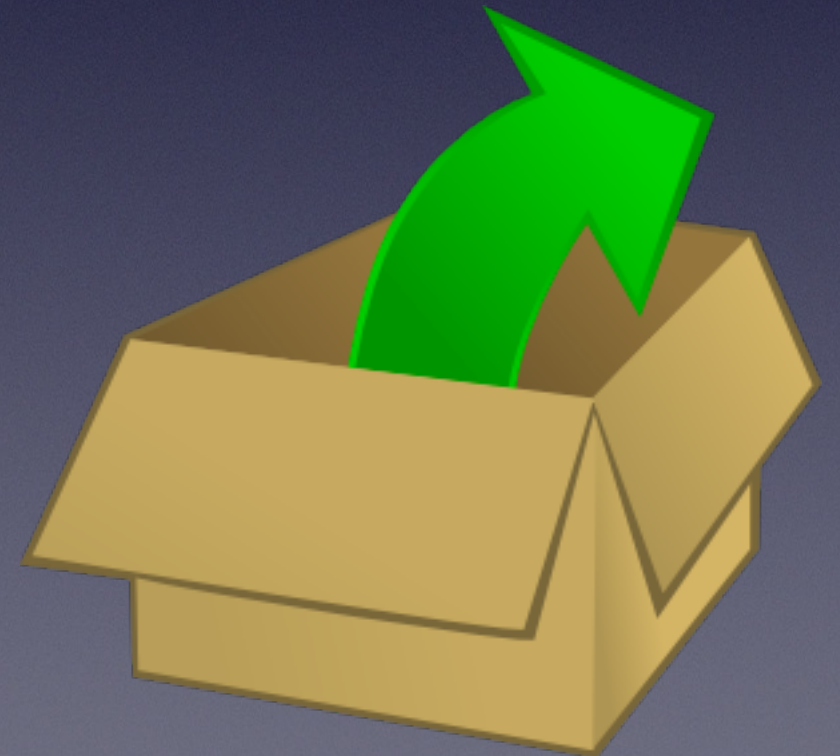
Review Board

- Web-based interface

- Commercially maintained open-source tool

- One trivial change to the authentication system to prevent students from modifying their submissions

Management Tool

- Abstracts away low-level commands to classroom commands like:
  - create-homework
  - distribute-homework
  - upload-reviews
- Keeps single central list of registered students

OS @ Columbia

- 6 two-week programming assignments

- Modifying the Linux kernel for Android

- Intro to C, system calls, synchronization, virtual memory, scheduler, file systems

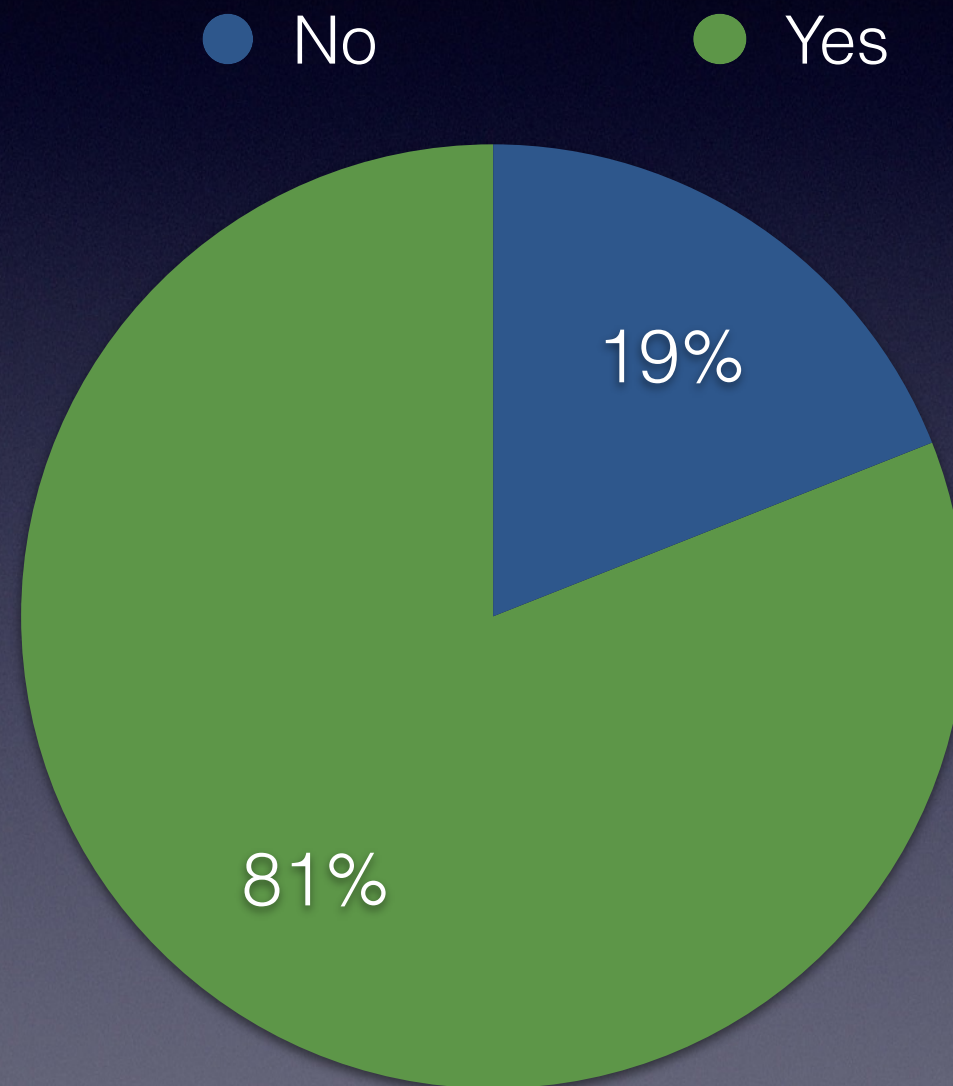- Live demonstrations

- Careful code review

Evaluation

- Used GradeBoard at Fall 2011 Columbia OS course

- Over 100 registered students
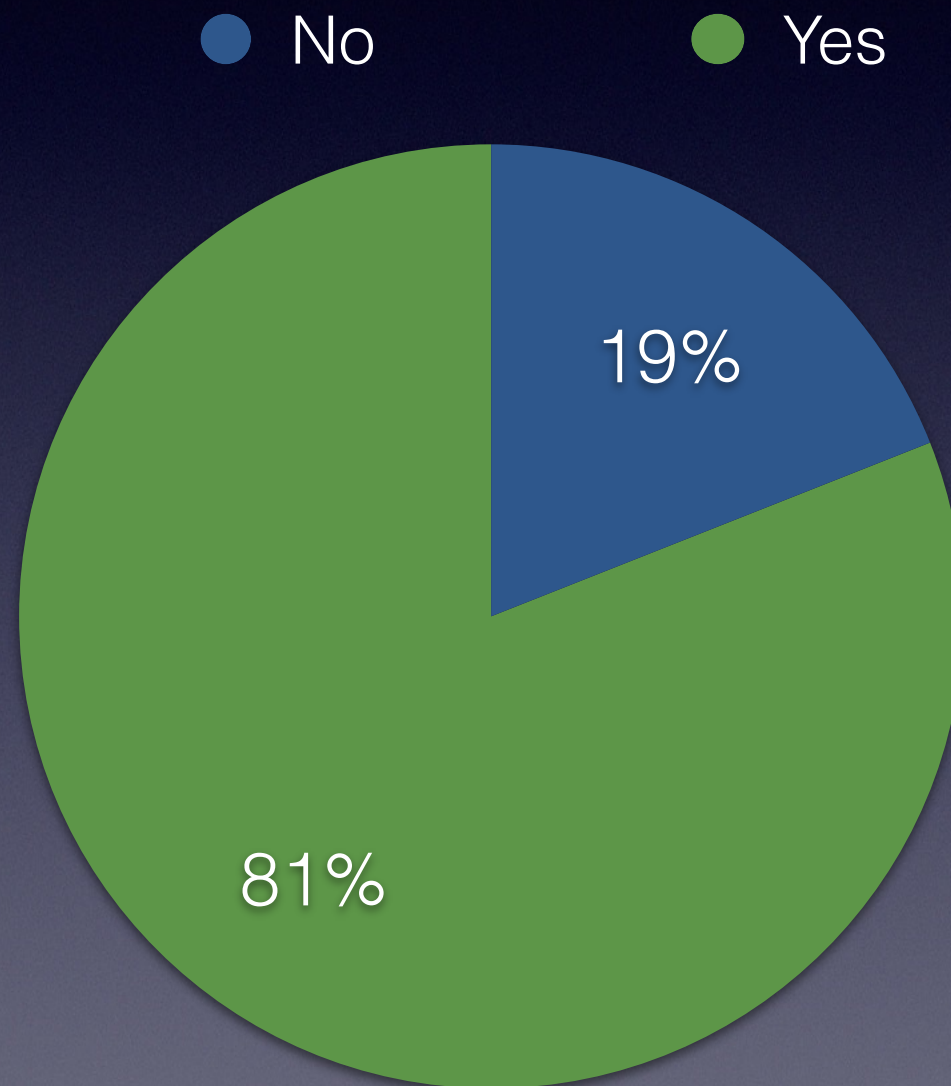
- Survey participation: 57%
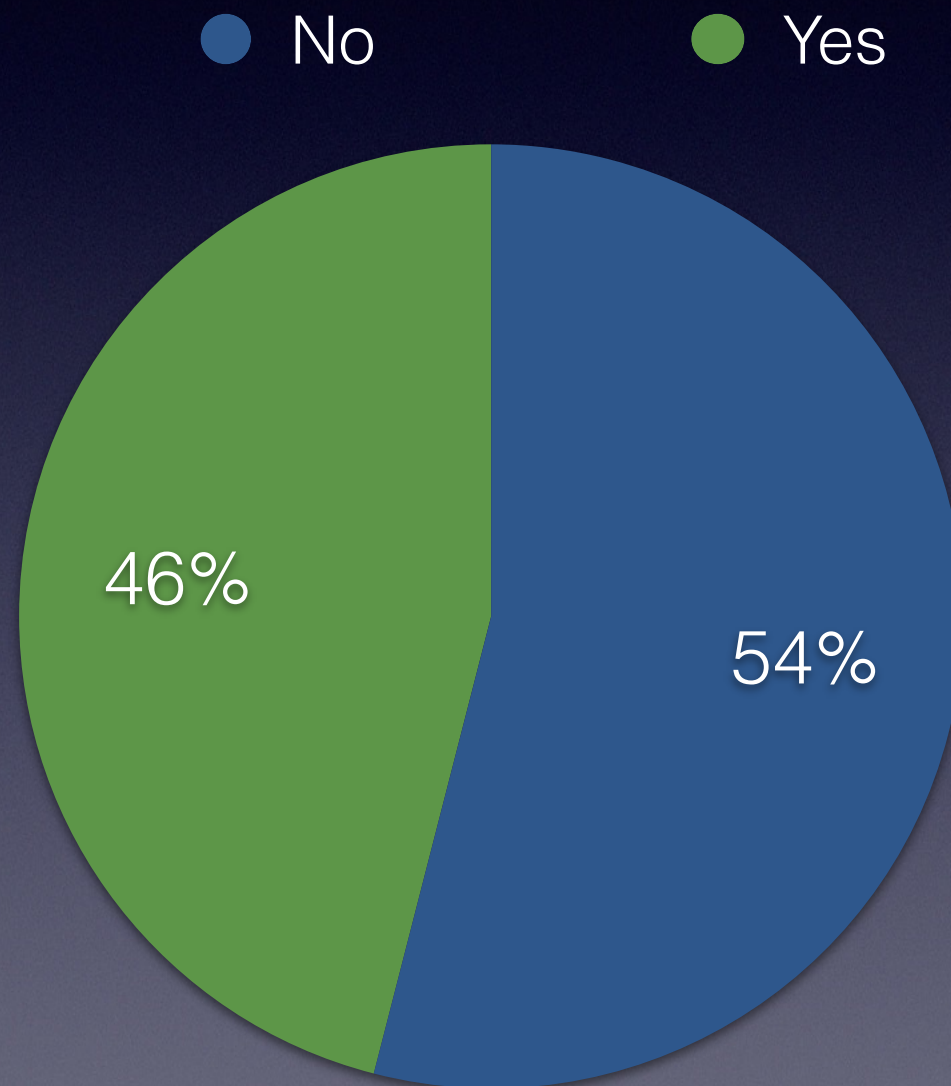
GradeBoard Improved Homework Understanding

No   Yes

8%

92%
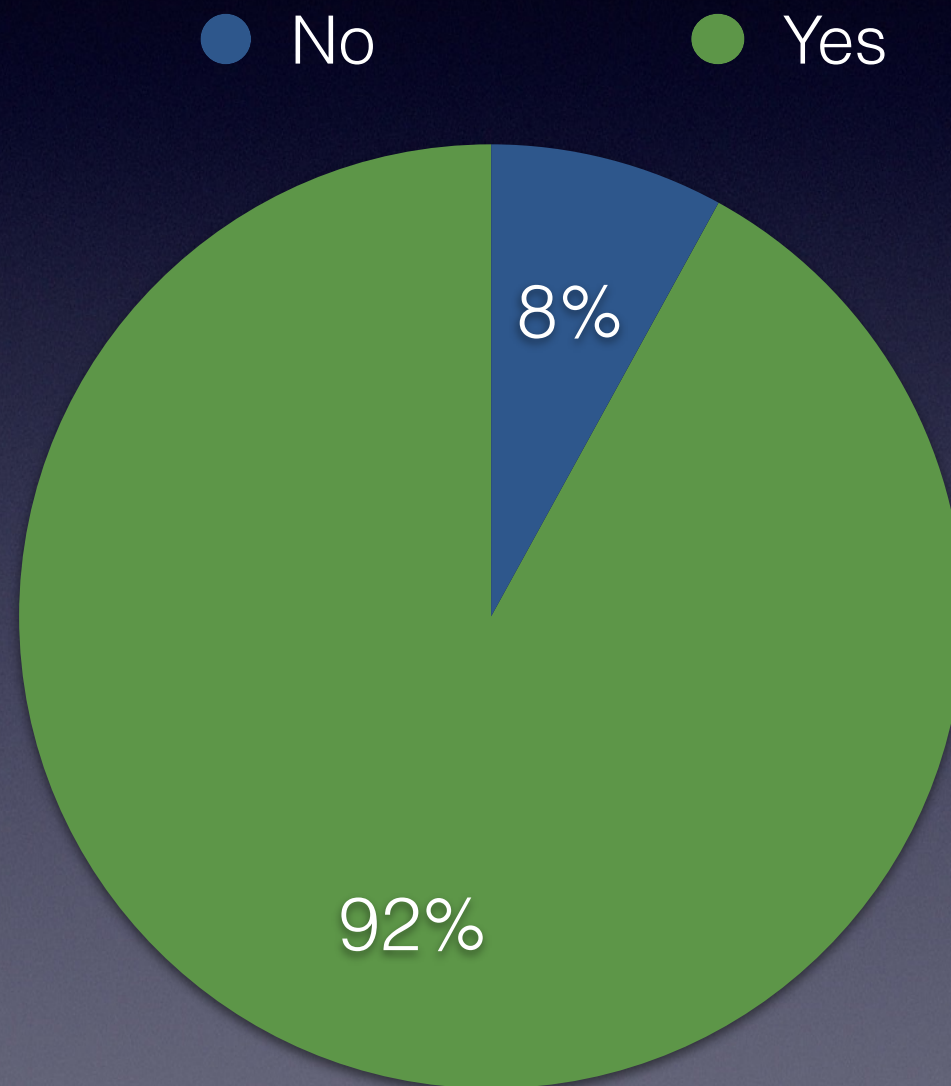
Preferred compared to e-mailed score sheets

- No
- Yes

19%

81%

Preferred compared to comments in Git

Encouraged well-formatted code

"I wish I had this tool when I taught operating systems last year."

–Nicolas Viennot, Head TA Fall 2009

Unexpected Benefit

Conclusions

- Implemented and evaluated GradeBoard

- Students learn better with GradeBoard for 2 reasons:
  - Comments inlined with student work
  - Student feedback identifies problem areas

- Over 80% of all submitted projects were discussed

- Free and maintained

- Students and instructors prefer GradeBoard over known alternatives

Questions?

Backup Slides…

Potential Solutions

- Grade Sheets: Manual and tedious

- Inline comments: requires use of command-line tools, time consuming

- Difficult for students to use

Problem Space

- Most hands-on OS assignments are based on some existing code base

- Both teaching OSes and commercial OSes

- No read-from-start-to-finish

Problem Space: Options

- Live demonstrations

- Automated Testing

- Doesn't work, must evaluate the code