AGILE COMMUNICATORS: Cognitive Apprenticeship to prepare students for communication-intensive software development

This work is supported by the National Science Foundation under Grant #1504860. Additional support from Jackson Blended Learning, Google CS Engagement.
Who we are
Who we are

Undergraduate programs: CS, SE (~350 students), plus MS, PhD
Strong reputation for preparing students for the workplace
In Week 1, Hank had asked the Software Engineering team to analyze some of his existing controller code:

Hank: Last semester, we kept trying to define how to proceed, and I kept suggesting that we make a skeleton of the code, and how I did that was by writing the names of the blocks and what they did in the code. And that is the kind of thing I would hope to happen here and how you want to organize the code.

Now it’s Week 6. Denise has done the code breakdown and has arranged a meeting with Hank to discuss it.
Seabase: Denise’s diagram

"unprofessional"  "embarrassing"
Dr. Taylor: Do you know from that [the chart] what values you will need?

Denise: This is where we need some help. So this is what happens in the code [pointing at Denise's chart]

Dr. Taylor: Can you show me some example within the code? This is great. Don't throw this out. Is this hand-drawn?

Dr. Taylor: Oh that is sweet! That makes sense now. So when this one is high, that value becomes high and this one goes low, that value is low. I finally get it.

Dr. Taylor: Wow! I love it! The beauty of something like this is that I can understand it. Someone with a high level of knowledge of how the code or the function works can look at it and completely understand it.
Denise’s communication challenges

She must think both strategically and tactically about the current problem at hand: the **stakeholders** involved, **when and whether** communication can help, and if it can, the **form** of communication that will solve it most effectively.
Placing communication in the curriculum

Writing and communication studies experts have a role ... But CS/SE instructors must step up too

- Grounding the material in authentic practices
- By attending to communication, we validate it

Expanded treatment of communication skills in SWEBOK 3.0

- Proficiency in genres
- Communication as **strategic** endeavor
What is agile communication?

**Proactivity:** Agile developers must be unafraid to inquire about requirements, to critique design choices, and to provide reflective comments on the team’s process.

**Flexibility:** Agile developers must be able to handle multimodal discourse (including written, oral and graphical communication through various media) and adapt to new communication situations, instead of relying on formal scripts and templates.

**Creativity:** Agile developers must be skilled rhetoricians, with a deep understanding of their communication options, and an ability to choose genre and style to suit the audience and purpose.
Cognitive apprenticeship

Engaging students in **authentic** software settings

**Articulating** processes that are traditionally left implicit

Students internalize agile practices as part of their personal software development process

Communication-intensive activities woven through the Computer Science curriculum

(Extreme Apprenticeship: Vihavainen & Luukkainen)
Communication-mediated activities

**Inquiry**: continuous interaction with stakeholders to understand ever-changing requirements, priorities and obstacles. In a process of continual self-optimization, teams self-organize and solve problems together.

**Critique**: refactoring when changing user needs and design demands dictate. Likewise, team organization and practices are also under constant review. This requires developers to be willing and able to reassess current design and practices and to articulate areas of improvement.

**Reflection**: A key component of agile development is continual process improvement, facilitated by periodic reflective activities.
Carefully crafted questions lead students through this cycle

(CS-POGIL: Kussmaul & Hu)
Process Oriented Guided Inquiry Learning

Attention to process

- Small self-managed teams, well-defined roles
- Authentic; encourages accountability and engagement

Team Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recorder</td>
<td>Records all answers &amp; questions, and provides copies to team &amp; facilitator.</td>
</tr>
<tr>
<td>Speaker</td>
<td>Talks to facilitator and other teams.</td>
</tr>
<tr>
<td>Manager</td>
<td>Keeps track of time and makes sure everyone contributes appropriately.</td>
</tr>
<tr>
<td>Reflector</td>
<td>Considers how the team could work and learn more effectively. Puts together a statement about how the team operated and characterize it as a communication pattern. Notice the interactions.</td>
</tr>
</tbody>
</table>
## Communication pattern rubric

<table>
<thead>
<tr>
<th>Category</th>
<th>Attribute</th>
<th>Possible Values/Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>How</strong></td>
<td>Style</td>
<td>Is the tone of the communication formal or informal? Is there a predefined structure in place for the communication, or is the structure to be defined during the communication itself?</td>
</tr>
<tr>
<td></td>
<td>Use of artifacts</td>
<td>Are there artifacts (tools like written material, diagrams, code) involved? Are they physical or virtual? Who owns them? Who has access to them?</td>
</tr>
<tr>
<td><strong>Where</strong></td>
<td>Location</td>
<td>Is there a particular place where the communication takes place? Or does it happen virtually, in no particular place? What attributes of the location are important for the communication to be effective?</td>
</tr>
<tr>
<td><strong>When</strong></td>
<td>Duration</td>
<td>Is there a fixed or typical duration for the communication act?</td>
</tr>
<tr>
<td></td>
<td>Synchrony</td>
<td>Is the communication mode synchronous (with instant response) or asynchronous (with no expectation of response time)?</td>
</tr>
<tr>
<td></td>
<td>Frequency</td>
<td>Is there an expected or common frequency for this communication act – once a project lifetime, weekly, daily?</td>
</tr>
</tbody>
</table>
Standup meeting

**Exploration**: Identify communication attributes (where/when/...)

**Invention**: Standup pattern as a combination of these communication attributes

**Application**: Consider the following “tweaks” to the standup pattern. How do they affect their use and relevance?

- **WHO**: all possible stakeholders in the project.
- **WHEN**: twice in the whole project life cycle.
- **WHEN**: half day in duration.
Communication patterns

Increasing complexity/authenticity; removing scaffolding

- Established agile practices
- Case studies from student projects
- Case studies from industry projects
- Students’ own experiences
Learning through engagement

<table>
<thead>
<tr>
<th>With others</th>
<th>With ourselves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project-external communication</td>
<td>Internal communication</td>
</tr>
<tr>
<td>Analyzing Scrum practices</td>
<td>Sprint reflection</td>
</tr>
<tr>
<td>Student team communication</td>
<td>Individual peer evaluation</td>
</tr>
<tr>
<td>Distributed collaborative teams</td>
<td>How we scrum / How they scrum</td>
</tr>
<tr>
<td>User interviews</td>
<td>Standup assessment</td>
</tr>
</tbody>
</table>

- having to recount specific situations
- general situations, more expanded
- more complex forms - simulated conflict

- more role-playing
- more practice in articulating problems & constraints

- simple & obvious
Reflection: How We Scrum

“For the Q&A sessions, the positive attribute is the **synchrony** of the session. The downside is the **focus of the activity** – as it can be lost easily. We can fix this issue by encouraging individuals to tell the team when they are sidetracked. For the **Google Docs spreadsheet**, the positive attribute is shared knowledge – every team member has access to the sheet, and can see tasks. The negative is the **frequency** – updates only really occur once per sprint, and sometimes once a week. To address this, we now plan on checking the spreadsheet **daily**.”
“Our communication style and format is largely based on face to face meetings several times a week whereas the other team depends much more on google docs to keep each other up to date. This face to face format keeps the frequency of our communications to a set number and time/place whereas the other team is in asynchronous communication constantly. For example, we find it beneficial to meet and discuss how to approach a problem, divide up the work, and then part ways to work on it separately, coming together again for our next meeting to discuss our progress. We find this helps keep each other accountable for the work that needs to be done. In contrast the other team does this much less, but updates google docs much more frequently. This allows them to spend whatever time they have available on a given task and work more independently.”
We teach students test-driven agile methods through small cycles of teaching, coding integrated with testing, and immediate feedback.
Students are repeatedly exposed to patterns of coding and critiques, from which they learn how to identify and communicate about issues that crop up during software development.
WebTA: Automated iterative code critique

Students reflect on just-in-time code critiques and adjust their coding style in a continuous development process.
WebTA: Automated iterative code critique

Critique Summary

- Compilation succeeded!
- There were 1 warnings, which could indicate subtle problems in the code.
- You passed all the instructor tests.
  - You passed all the secret tests.
  - You passed some student tests, but failed others.
- You passed some reverse tests, but failed others.
- I found some style issues in your code. See below.

Print
WebTA: Automated iterative code critique

**Test:** fibRecursive Test.
**Description:** Checks for all values of n from 0 to 9.
**Hint:** Remember to test for the base case.
**Status:** Failed

- java.lang.StackOverflowError
- Fibonacci.fibRecursive(Fibonacci.java:17)
- Fibonacci.fibRecursive(Fibonacci.java:20)
- Fibonacci.fibRecursive(Fibonacci.java:20)
- Fibonacci.fibRecursive(Fibonacci.java:20)
WebTA: Automated iterative code critique

```java
for (int i = 0; i < n; i++) {

Style Critique
Status: Warning
```
Lessons learned (so far)

Instructors
• Prolonged, intensive involvement with communication is key
• POGIL structure helped focus students, elicit more substantial answers
• Written responses to POGIL activities not as rich as class discussion
• With POGIL preparation, students reflect in a rhetorically sophisticated way
• Students aren’t hostile to discussing communication if they see that it helps

Students
• Regular reflection, analysis of activities improved process
• Did not need to revamp their communication strategies in the final sprint as they had reflected and improved frequently in the first two.
WebTA: Future developments

In our targeted courses, we have observed “divergent” student coding behavior that does not reflect good practice despite passing automated tests. Based on these observations, we are formulating common **coding antipatterns** and devising **heuristic triggers** within WebTA that detect these behaviors.

To facilitate enhanced communication, reinforce reflection, and support lab-based pair-programming practices, we are developing an online **pair programming development environment** that will be tightly integrated with WebTA. This larger system will enable us to detect and provide feedback on **student communication patterns** and development practices that we cannot observe by doing a post-facto analysis of code in WebTA.
AGILE COMMUNICATORS: Cognitive Apprenticeship to prepare students for communication-intensive software development

This work is supported by the National Science Foundation under Grant #1504860. Additional support from Jackson Blended Learning, Google CS Engagement.

Shreya Kumar
shreyaK@mtu.edu
@shreyakumar7

Leo Ureel
ureel@mtu.edu
@MTU_Leo

Charles Wallace
wallace@mtu.edu
@CRWallace3