**Feedback comment:**
Include coverage of logging - volume of logs, logging in production, thread safety, timestamps, etc.

**How incorporated:**
Added as a potential KA-Core topic in Software Construction

**Why not incorporated:**

---

**Feedback comment:**
Include coverage of asyncrony

**How incorporated:**

**Why not incorporated:**
Out of scope, handled in parallel and distributed computing

**Date considered:** Jun 15, 2023

---

**Feedback comment:**
Include coverage of DevOps

**How incorporated:**

**Why not incorporated:**
Added as a relevant Non-core topic in Software Construction

**Date considered:** Jun 15, 2023

---

**Feedback comment:**
Requirements engineering should be more emphasized.

**How incorporated:**
**Why not incorporated:**

SE committee has already discussed relative priority of these knowledge units, we don’t believe that product requirements is *more* essential than the existing CS Core topics. Product requirements is already represented in KA Core.

**Date considered:** Jun 15, 2023

---

**Feedback comment:**

Include coverage of software security

**How incorporated:**

---

**Why not incorporated:**

Out of scope, handled in SEC. There are already several cross-listings between these areas.

**Date considered:** Jun 15, 2023

---

**Feedback comment:**

Replace pair programming with swarming

**How incorporated:**

Added swarming as an alternative to pair programming in the Teamwork KU

---

**Why not incorporated:**

---

**Feedback comment:**

Include greater coverage of client communication

**How incorporated:**

---

**Why not incorporated:**

Has been previously discussed, isn’t believed by the committee to be more relevant to new grads than the existing topics.
Date considered: Jun 15, 2023

Feedback comment:
Drop Software Processes - Companies already mandate this and textbooks aren’t good on the material

How incorporated:

Why not incorporated:
The intent is to provide students with an awareness of the types of processes they are likely to encounter, specifically to prepare them for their first days on the job. They aren’t expected to master these things, but to come away from this KA with some awareness of what to expect.

Date considered: Jun 15, 2023

Feedback comment:
Drop Formal Methods

How incorporated:

Why not incorporated:
Previously discussed by the committee, no consensus to drop. Already limited to non-core requirement.

Date considered: June 15, 2023

Feedback comment:
Broadly - strengthen the learning outcomes.

How incorporated:

Why not incorporated:
Given the limited CS Core budget, reaching an “apply” level of mastery for these outcomes seems infeasible. Aiming for a conceptual understanding in order to prepare students for their first position post-graduation is the best we can do.

Date considered: June 15, 2023
**Feedback comment:**
Move “version control” from tools to teamwork

**How incorporated:**
Cross listed “version control” in both KUs (adding to “Teamwork”)

**Why not incorporated:**

---

**Date considered:** June 15, 2023

---

**Feedback comment:**
Incorporate REST and microservices in Software Design

**How incorporated:**
Made explicit in the KU for Software Design, alongside existing client-server mention.

**Why not incorporated:**

---

**Date considered:** June 15, 2023

---

**Feedback comment:**
I don’t see anything about ethics

**How incorporated:**

---

**Why not incorporated:**
Specifically covered in the SEP knowledge area (with appropriate cross-listings).

**Date considered:** June 15, 2023

---

**Feedback comment:**
The importance of documentation and related Standards are missing.

**How incorporated:**

---

**Why not incorporated:**
Communication generally and commenting specifically are already covered. Higher-level documentation is valuable as well, but less well defined and potentially less relevant for new-grads.

**Date considered:** June 15, 2023

---

**Feedback comment:**

Agile methodology is not suitable for many (large-scale) projects! Hence limitations of each methodology and their applicability to variety of project types need to be brought out.

**How incorporated:**

**Why not incorporated:**

Irrelevant? Agile is barely highlighted (primarily at the request of previous reviewers). General conceptual introduction to types of team process is what we are aiming for.

**Date considered:** June 15, 2023

---

**Feedback comment:**

Pitfalls for Teamwork and hence the need for developing a realistic Hurt Statement/s need to be emphasized.

**How incorporated:**

**Why not incorporated:**

Generically covered in “Common causes of team conflict, and approaches for conflict resolution”, but we are not mandating specific approaches.

**Date considered:** Jun 15, 2023

---

**Feedback comment:**

Intellectual Property (IP) management in Software Engineering needs to be emphasized.

**How incorporated:**

**Why not incorporated:**

Out of scope for new-grads, which is the target audience. Some conceptual coverage of IP generally already exists in the SEP KA, but specific approaches to managing IP are out of scope.
Date considered: June 15, 2023

Feedback comment:
Importance of learning elements from Sociology, Psychology & Anthropology and their relevance to Software Engineering needs to be brought out. Similarly, the relevance of fields such as management, accounting, etc, to Software Engineering need to be brought out.

How incorporated:

Why not incorporated:
Unclear how to incorporate? Teamwork seems like it ought to cover some of this. Similarly, SEP likely has some overlap?

Date considered: June 15, 2023

Feedback comment:
Importance and relevance of use & deployment of COTS (commercial off-the-shelf) products in real-life projects needs to be brought out. COTS rule the real-world software systems; no one develops software from scratch.

How incorporated:

Why not incorporated:
We are mostly agnostic on both OSS and COTS. It is unclear how much background a new-grad must have as a minimum requirement (which is the role of this document).

Date considered: June 15, 2023

Feedback comment:
Component Based Software Engineering (CBSE) is widely used/deployed in industry. Tools and techniques in this arena must be taught.

How incorporated:

Why not incorporated:
It is unclear how much background a new-grad must have as a minimum requirement (which is the role of this document).
Feedback comment:
Newer type of software costing models for systems using/deploying CBSE, COTS, etc must be inducted.

How incorporated:

Why not incorporated:
Not relevant for a new grad.

Date considered: Jun 15, 2023

Feedback comment:
The use of various Software Standards (from IEEE, ISO, ANSI) in various fields (e.g., ISO108, ISO184) has to be made relevant in the study of software engineering.

How incorporated:

Why not incorporated:
Not relevant for a new grad.

Date considered: Jun 15, 2023

Feedback comment:
API testing using algebraic formulation (e.g., event algebra and sequence algebra) needs to be brought out.

How incorporated:

Why not incorporated:
We couldn’t find references to these terms, much less convince the sub-committee that these were critical as a minimum requirement for graduates.

Date considered: Jun 15, 2023
Feedback comment:
Software maintenance takes up 60% of the overall life-cycle cost of software systems (according to EDS, InfoSys and TCS). This area needs to be emphasized and taught elaborately as the biggest companies are no longer software developing companies, but instead software maintenance companies.

How incorporated:

Why not incorporated:
The committee has already discussed the relative weight of knowledge units - this is covered (not by the same terms) in Refactoring and Evolution, as well as general SDF requirements for code reading/understanding. We agree that this is valuable, but the balance presented (some KA-Core coverage) is the best we believe we can manage given limited lecture hours and the practicalities of undergraduate education.

Date considered: Jun 15, 2023

Feedback comment:
Training is something forgotten, but it consumes considerable time and effort. How to train users is an important topic to consider (and hence the writing of appropriately worded user manuals).

How incorporated:

Why not incorporated:
Not relevant for a new grad.

Date considered: Jun 15, 2023

Feedback comment:
New areas such as Software Culture needs to be inducted – i.e., how one gets indoctrinated into a company software architecture, designs, management, processes and procedures and cannot get out easily. More details & references on this topic can be provided.

How incorporated:

Why not incorporated:
Would classroom coverage of this address the problem at all? This seems distinctly like a topic that is primarily learned through experience, and that is particularly ill-suited to the classroom.

**Date considered:** Jun 15, 2023

---

**Feedback comment:**
The deployment and use of AI tools (e.g., GPT4, etc) for automating coding, configuration management, version control, team management, etc, have to be inducted.

**How incorporated:**
Added as an exemplar in Tools and Environments.

**Why not incorporated:**

---

**Date considered:** Jun 15, 2023

---

**Feedback comment:**
Large-scale software systems & their visualization and maintenance is becoming an important issue as software systems become larger and larger over time.

**How incorporated:**

**Why not incorporated:**
Far easier to handle on the job than in the classroom.

**Date considered:** Jun 15, 2023

---

**Feedback comment:**
Provenance management in software systems is/will become important. This is another upcoming area. More details & references on this topic can be provided.

**How incorporated:**

**Why not incorporated:**
Not relevant for a new-grad.

**Date considered:** Jun 15, 2023
Feedback comment:
we could also introduce the concept of capstone project courses, with industry sponsors, real projects, etc.

How incorporated:
Added to the preamble.

Why not incorporated:

Date considered: Jun 15, 2023

Feedback comment:
In my opinion project management is still a key aspect of software engineering projects. Just to give an example, at UCI the master of Software Engineering program offers in the last quarter, 3 soft skills courses: project management, career & entrepreneurship, capstone projects. I am not confident that we should remove PM from the knowledge units. And I would seriously consider adding something around problem solving / entrepreneurial mindset applied to SE. Just some out of the box ideas.

How incorporated:

Why not incorporated:
Out of scope for undergraduates.

Date considered: Jun 15, 2023

Feedback comment:
What about forming and developing effective teams?

How incorporated:

Why not incorporated:
Out of scope for undergraduates.

Date considered: Jun 15, 2023
Feedback comment:
Should we be explicit about oral and written communication requirements?

How incorporated:
Made explicit in “Effective communication” in the Teamwork KU

Why not incorporated:

Date considered: Jun 15, 2023

Feedback comment:
Conflict resolution and also problem solving

How incorporated:
Expanded reference to “conflict resolution” in the learning outcomes for Teamwork to include problem solving.

Why not incorporated:

Date considered: Jun 15, 2023

Feedback comment:
What about requirements engineering tools?

How incorporated:

Why not incorporated:
Believed to be out of scope for new grads.

Date considered: Jun 15, 2023

Feedback comment:
What about mentioning technical debt as a possible impact of not having refactoring and code evolution?
How incorporated:
Mentioned technical debt as a potential bad outcome without techniques in refactoring and evolution.

Why not incorporated:

Date considered: Jun 15, 2023

Feedback comment:
"Importance of team diversity and inclusivity"

How incorporated:
Expanded the topic listing in the Teamwork KU accordingly

Why not incorporated:

Date considered: Jun 15, 2023

Feedback comment:
I wonder whether some topics from Design Thinking/Lean Startup are relevant here, for example, customer interview, empathy, personas?

How incorporated:

Why not incorporated:
Those are great texts for early/mid-career professionals, but generally out of scope for new-grads.

Date considered: Jun 15, 2023

Feedback comment:
The Dave Parnas quote in the SE preamble is actually Parnas quoting Brian Randell.

How incorporated:
Changed attribution.

Why not incorporated:
**Date considered: Jun 15, 2023**

---

**Feedback comment:**

It’s not the role of an undergraduate programme to work on team structures, development methodologies, project planning, specification writing, project estimation

**How incorporated:**

**Why not incorporated:**

Broadly agreed, these are intended only for light (and generally optional) coverage. We believe the document already reflects this suggestion.

**Date considered: Jun 15, 2023**

---

**Feedback comment:**

Minor survey contention on the question of whether Software Design is an essential topic.

**How incorporated:**

**Why not incorporated:**

Not enough dissent to re-evaluate, still an overwhelming plurality of support. The required level of coverage is also quite minimal - we want to hint at the ideas that graduates are likely to learn on the job, but are unlikely to be able to instill mastery.

**Date considered: Jun 15, 2023**

---

**Feedback comment:**

Doesn’t address influences and resulting approaches to overall development: prioritizing development actions based on perspectives of the process characteristics, linear, iterative, holistic, risk-based, time-to-market critical, proof-of-concept, etc.

**How incorporated:**

**Why not incorporated:**

Likely not relevant for new-grads.

**Date considered: Jun 15, 2023**
Feedback comment:
Some survey contention on the question of whether graduates should be aware of stylistic differences between tests and production code (i.e., DAMP vs. DRY)

How incorporated:

Why not incorporated:
Executive decision from Titus Winters on this one - tests are untested and need to be correct by inspection. If dataflow is obscured or the correctness/validity of a test is non-obvious, the value of the test drops dramatically. DRY is valuable for code maintenance, but the priorities in test code are distinct.

Date considered: Jun 15, 2023

Feedback comment:
Introduction to formal methods at a basic undergraduate level

How incorporated:

Why not incorporated:
Already discussed repeatedly, no new information

Date considered: Jun 15, 2023