

# Product Backlog „Saros“

## BEREICH „AWARENESS-VERBESSERUNG“

ID	Beschreibung	Priorität	Status
NiceCursor	Die Text-Cursor anderer Sitzungsteilnehmer sollen auch als Cursor angezeigt werden, nicht nur als farbige Box (denn diese ermöglicht keine Unterscheidung zwischen leerer und einer Ein-Zeichen-Selektion). Optional: Kleiner Marker am Cursor mit Name und Farbe des jeweiligen Nutzers.	mittel	offen
HighlightMirror	Die lokale Hervorhebung von Bezeichnerverwendungen nach einem Anklicken (nicht Selektion) des Bezeichners soll auch für die anderen Sitzungsteilnehmer sichtbar sein.	hoch	offen
ContribFade	Aktuell richtet sich die Anzeigedauer der sog. „Contribution-Annotations“ anderer Sitzungsteilnehmer nach Anzahl der Änderungen. Gewünscht ist ein Ausblenden älterer Annotation nach Verstreichen einer bestimmten Zeitspanne. Optional: Statt einem harten Verschwinden sollten die Annotation sanft ausgeblendet werden.	mittel	offen
AutoCompMirror	Die Autovervollständigungsboxen sollen ebenfalls für die anderen Teilnehmer sichtbar sein.	mittel	offen
BulkIgnore	Massenhafte automatische Änderungen z.B. der Einrückung (etwa durch automatische Formatierung beim Speichern) sollen nicht als „Contribution“ angezeigt werden, die tatsächliche Textänderungen überschatten.	mittel	offen
SelectAnnoFast	Die aktuell als „experimentell“ gekennzeichnete blockige „Selection-Annotation“ ist für den produktiven Einsatz deutlich zu langsam.	mittel	offen

### Verfahrenshinweise

#### Punkt „NiceCursor“

Im Jahr 2013 wurde ein solches einmal begonnen. Status des Features ist unklar. Viel Entwicklungsarbeit wird hierbei aber nicht mehr zu leisten sein.

#### Punkt „AutoCompMirror“

Für die weitere Evaluation ist es zunächst ausreichend das Feature nur im sog. „FollowMode“ zu aktivieren.

## BEREICH „SAROS FÜR INTELLIJ“

ID	Beschreibung	Priorität	Status
IntComp	Saros für IntelliJ soll die aktuelle Version 2016.2 unterstützen, nicht nur die veraltete Version 13.	mittel	offen
IntOverview	Für die Planung der weiteren Entwicklung brauchen wir eine Übersicht über die <i>Mängel</i> und <i>Funktionslücken</i> in der aktuellen Implementierung von Saros für IntelliJ.	hoch	offen

### Verfahrenshinweise

#### Punkt „IntComp“

Es könnte evtl. von Vorteil sein, die Versionsupgrades Schritt für Schritt durchzuführen, d.h. erst von Version 13 auf 14, dann auf 15 und dann auf 2016.2 zu gehen, um möglichst wenige API-Änderungen auf einmal behandeln zu müssen.

Letztlich muss auch die Build-Server-Infrastruktur des Saros-Projektes angepasst werden, die aktuell auf IntelliJ in Version 13 ausgelegt ist.