

Feedback voorlopige analyse 2022

Algemene tips

- Het eindverslag liefst indienen in pdf-vorm, met een bestandsnaam waarin ook het groepsnummer zit. Op die manier kunnen uw begeleiders dit makkelijkst verbeteren op een e-reader.
- Maak voor het eindverslag een net voorblad met vermelding van naam van het opleidingsonderdeel (Softwareproject), titel die de lading van jullie project dekt, groepsnummer, namen van teamleden, jaargang, en logo's van UGent en de faculteit (zie later meer). Geen paginanummer of footers op een voorblad!
- Vanaf bladzijde 3 vinden jullie feedback specifiek voor de analyse van jullie groep. Deze analyse stond *niet* op punten. De feedback dient enkel om het eindverslag proactief te verbeteren. En om jullie op weg te zetten voor de verdere implementatie van het project. Lees elke opmerking van jullie coaches als een *bedenking*. Als jullie er toch anders over *denken*, dan mogen jullie met goede reden uiteraard afwijken van onze tips.
- De diagrammen werden met de hand becommentarieerd. Deze aanvullingen zijn te vinden op een kopie van jullie pdf, in dezelfde map *feedback_door_docenten* als waarin dit bestand staat.

Misschien zie je een paar van onderstaande tips nogmaals verschijnen in het lijstje op de volgende bladzijde. Dan was deze tip dus zeker voor jullie groep bedoeld ☺

- Sommige groepen gebruiken nog het oude logo van de faculteit. Voor de meest recente versie, zoek met de termen 'ugent logo' en je vindt een site waar de juiste logo's afgehaald kunnen worden. Let op: de kleur van het nieuwe faculteitslogo is recent veranderd, dus haal het zeker van de site.
- Bij het aanmaken van objecten in het sequentiediagram staan de afbeeldingen voor het object (de rechthoekjes) lager, namelijk op de plaats waar ze aangemaakt worden. Kan dat niet met de tool die je gebruikt? Zoek er even op, anders laat je het ons weten. Maar maak in elk geval met een horizontale pijl met opschrift *new...* duidelijk welke klasse verantwoordelijk is voor het aanmaken van welke objecten! (En worden er meerdere objecten aangemaakt, gebruik dan geen gewoon rechthoekje maar een rechthoekje met daarachter in perspectief nog een tweede rechthoekje.)
- Bij het beschrijven van de werking van een GUI in een sequentiediagram mag je html- en script-code als één geheel beschouwen. Want de koppeling tussen tags in de html en bijhorende data of methodes in het script verloopt automatisch.

- In een wetenschappelijke tekst worden alle namen van variabelen, klassen, grootheden, (stukken van) formules of code in een ander lettertype geschreven. Neem een half voorbeeld aan deze tekst: hier werd het af en toe gedaan (vergelijk de leesbaarheid met de stukken waar het niet gedaan werd). Gebruik voor code bij voorkeur een typewriter-lettertype, waar elke letter evenveel plaats in de breedte inneemt.
- Begin een zin NOOIT met de naam van een variabele, klasse, of methode. Want ten eerste kom je dan in de problemen met de naamgevingsconventies (hoofdletter of niet?) en ten tweede is het onduidelijk. **De oplossing:** als de zin met de naam van een variabele, methode,... start, voeg dan de soortnaam van dat ‘ding’ toe. Een voorbeeld uit een wiskundige context:

l is een rechte die gaat door P.	De rechte l gaat door het punt P.
P is een punt dat ligt op l.	Het punt P ligt op de rechte l.

En in een informatica-context:

Persoon heeft een constructor zonder parameters.	De klasse Persoon heeft een constructor zonder parameters.
janneman heeft schoenmaat 28.	Het object janneman heeft schoenmaat 28.
zing(liedje) is een logische methode die teruggeeft of er vals gezongen werd.	De logische methode zing(liedje) geeft terug of er vals gezonden werd.

- Scan op het einde de volledige tekst op het gebruik van *ons*, *onze*, *we*, *je*, *jij*, *jou(w)*, *jullie*,... : dat vermijd je ☺ in een vormelijke tekst.

Logboeken

Jullie krijgen een screenshot van de huidige stand van de logboeken. Kijk dat na op git. Als er in het overzicht onderaan gele (of rode) blokken staan, wil dat zeggen dat tussen de huidige snapshot en de vorige snapshot geen nieuwe uren gelogd werden. Staat er **ERROR** in een rood blok, dan zat er ergens een foutje in de structuur van dat logboek, en is er dus geen informatie voor de snapshot van die dag. Graag nakijken!

Eventuele rode kaders met **Total Open / Total Closed** zijn niet van toepassing, daar hoeft je niet op te letten.

De blauwe lijn in de eerste afbeelding volgt het aantal uren dat aan het project gespendeerd zou moeten zijn in die week - om tot het eindtotaal te komen. De ‘paaspauze’ (het plateau in het rode blok) geeft jullie de kans om bij te benen!

Team-4 Tetris-bot (genetisch algoritme; data nog te bepalen?) *JavaScript*

- Geen analysedocument te vinden op git.
- Diagrammen in puml-formaat zijn niet door standaard software te lezen. Dus hoort hier wat uitleg bij of beter nog, een omzetting naar pdf. (Met bindteksten, zodat de lezer weet waar eerst gelezen.)
- In de readme: *which is used in various medical sectors as well* -> kunnen jullie hier een paar voorbeelden bij geven in het project? Het lijkt me interessant voor de gebruiker om ook wat achtergrondinformatie en uitleg te krijgen bij gebruik, ontstaan en werking van de categorie van genetische algoritmes.
- Misschien vind je een paar mooie citaten die verband houden met genetische algoritmes? Verwerk die in het project (en gebruik de juiste bronvermelding).
- Waar gaan jullie 'veel data' gebruiken in het project? Worden die gegenereerd, of ergens van het net gehaald? En kunnen jullie data van gespeelde spellen (en highscores) in grafieken trekken? In ieder geval de blokken die vallen uitlezen vanuit een bestand (of andere opslag).
- Gezien er voorlopig nog geen SDG aan het project gelinkt is, ga je voluit voor het educatieve! Dus duik samen met de gebruiker helemaal in het genetische algoritme (beter: in het concept van genetische algoritmes, waarvan je dan één specifieke toepassing als voorbeeld gebruikt).
- Leg jullie diagrammen terug voor aan de begeleiders, om ook daar feedback op te krijgen.