

MyTalk

Software di comunicazione tra utenti senza
requisiti di installazione



clockworkTeam7@gmail.com

Norme di Progetto

v 1.0

Informazioni sul documento

Nome documento	Norme di Progetto
Versione documento	v 1.0
Data creazione	28/11/2012
Data ultima modifica	03/12/2012
Uso documento	Interno
Redazione	Gavagnin Jessica Bain Giacomo
Verifica	-
Approvazione	Bain Giacomo
Lista distribuzione	gruppo <i>Clockwork</i> Prof. Tullio Vardanega

Sommario

Questo documento vuol definire le norme volte alla regolamentazione del gruppo *Clockwork* necessarie al processo di sviluppo del progetto **MyTalk**.

Diario delle modifiche

Autore	Modifica	Data	Versione
Bain Giacomo	Approvazione finale documento	03/12/2012	v 1.0
Gavagnin Jessica	Sistemazione e stesura capitoli 9 e 11	03/12/2012	v 0.9
Bain Giacomo	Bozza capitoli 9 e 11	30/11/2012	v 0.8
Gavagnin Jessica	Stesura capitoli 8 e 10	30/11/2012	v 0.7
Gavagnin Jessica	Sistemazione e stesura completa capitoli 4, 5, 6 e 7	30/11/2012	v 0.6
Bain Giacomo	Bozza capitoli 6 e 7	29/11/2012	v 0.5
Gavagnin Jessica	Creazione e stesura dei capitoli 1, 2 e 3	29/11/2012	v 0.4
Zohouri Haghian Pardis	Correzione bozza capitolo 5	29/11/2012	v 0.3
Ceseracciu Marco	Bozza del capitolo 4, repository	28/11/2012	v 0.2
Palmisano Maria Antonietta	Bozza del capitolo 5, sulla struttura dei documenti	28/11/2012	v 0.1

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Ambiguità	1
2	Comunicazioni	2
2.1	Comunicazioni interne	2
2.2	Comunicazioni esterne	2
3	Incontri	3
3.1	Incontri interni	3
3.1.1	Casi particolari	3
3.2	Incontri esterni	3
4	Repository	5
4.1	Cartella documenti	6
4.2	Cartella Codice	6
5	Documenti	7
5.1	Formattazione e copertina	7
5.2	Norme tipografiche	7
5.2.1	Stile del testo	8
5.2.2	Punteggiatura	8
5.2.3	Composizione del testo	8
5.2.4	Formati ricorrenti	9
5.3	Componenti visive	10
5.3.1	Immagini	10
5.4	Versionamento	10
5.5	Formattazione documenti	11
5.6	Tipi di documenti	12
5.6.1	Verbalì incontri	12
5.6.2	Documenti informali	13
5.6.3	Documenti formali	13
5.7	Procedura di formalizzazione dei documenti	14
5.8	Procedura di verifica dei documenti	14
5.9	Incompatibilità	14
5.10	Informazioni aggiuntive sui documenti	15
6	Analisi dei Requisiti	16
7	Studio di fattibilità	18
7.1	Rischi	18
7.2	Vantaggi	19
8	Progettazione	20

9	Codifica	22
9.1	Intestazione del file	22
9.2	Convenzioni di codifica	23
10	Glossario	24
10.1	Inserimento vocaboli	24
11	Gestione delle attività	25
12	Ambiente di progetto	27
12.1	Ambiente generale	27
12.1.1	Sistema operativo	27
12.2	Ambiente documentale	27
12.2.1	Scrittura documenti	27
12.2.2	Verifica ortografica	27
12.2.3	Pianificazione	27
12.2.4	Grafici UML	27
12.2.5	Mockup	28
12.2.6	Documentazione semi-automatica	28
12.3	Ambiente di sviluppo	28
12.3.1	Strumenti di versionamento	28
12.3.2	Ambiente di codifica	28
12.4	Ambiente di verifica e validazione	28
12.4.1	Analisi statica	28
12.4.2	Test	29
12.4.3	Analisi dinamica	29
12.4.4	Validazione	29

1 Introduzione

1.1 Scopo del documento

Il documento Norme di Progetto viene redatto per definire tutte le norme che regolamenteranno lo svolgimento del progetto. Gli argomenti che le norme trattano riguardano:

- *Relazioni interpersonali*
- *Redazione documenti*
- *Codifica*
- *Definizione delle particolarità dei vari documenti*
- *Definizione dell'ambiente di lavoro*

Tutti i membri sono obbligati a visionare questo documento ed a sottostare alle norme descritte, per poter migliorare l'efficienza, le operazioni di verifica, la coerenza e l'organicità tra i vari file prodotti.

Se si presenterà la necessità, ogni membro del gruppo *Clockwork* potrà suggerire cambiamenti o aggiungere eccezioni all'Amministratore di Progetto.

1.2 Ambiguità

Per evitare ogni ambiguità relativa al linguaggio e termini utilizzati nei documenti formali, viene allegato il glossario nel file `Glossario_v1.0.pdf`, dove vengono definiti e descritti tutti i termini che sono marcati da una sottolineatura.

2 Comunicazioni

2.1 Comunicazioni interne

Le comunicazioni interne avverranno esclusivamente tramite Facebook e il servizio di ticket offerto da GitHub, e verranno utilizzate nel seguente modo:

- **Nota su Facebook:** per comunicazioni rivolte a tutto il gruppo si lasciano delle note nel social network. Il messaggio dovrà iniziare con l'oggetto dello stesso nel seguente formato [OGGETTO]. Per rendere nota la visione dei messaggi, bisognerà commentare la nota
- **Ticket:** il servizio di ticket offerto da GitHub sarà usato per tutte le comunicazioni da persona a persona, ossia comunicazioni “point-to-point”. La natura del ticket si evidenzierà tramite l'apposizione all'oggetto dello stesso il tag [MESSAGGIO], senza indicare alcuna milestone

2.2 Comunicazioni esterne

Sarà il Responsabile di Progetto, che si occuperà di comunicare con persone esterne al gruppo, come il committente e il proponente.

A tale scopo è stata creata la casella mail *clockworkTeam7@gmail.com* tramite la quale il Responsabile di Progetto parlerà per conto e in nome dell'intero gruppo di progetto.

Sarà a cura del Responsabile di Progetto informare gli altri componenti del gruppo di eventuali corrispondenze pervenute da persone esterne al gruppo: in questo caso si applicheranno le norme relative alle comunicazioni interne definite nella sezione 2.1.

3 Incontri

3.1 Incontri interni

Gli incontri interni saranno fissati dal Responsabile di Progetto.

Tali incontri potranno essere indetti:

- Su richiesta di un membro del gruppo (tramite ticket indirizzato al Responsabile di Progetto) e con approvazione di almeno altri due membri del gruppo (ad esclusione del proponente stesso)
- Su proposta dell'Amministratore di Progetto e con l'assenso del Responsabile di Progetto o viceversa

Il Responsabile di Progetto dovrà occuparsi di comunicare tali incontri tramite un messaggio nel social network e dovrà contenere la **motivazione**, il **nome** del proponente, la **data**, l'**ora** e il **luogo**, esattamente in questo ordine. Tale comunicazione dovrà avvenire con almeno tre giorni di anticipo.

Ogni membro del gruppo ha la facoltà di fare richiesta di un incontro interno. Tale richiesta sarà indirizzata unicamente al Responsabile di Progetto, che la visionerà e pianificherà l'eventuale incontro.

Ogni membro dovrà confermare la sua presenza a tale incontro o che spieghi i motivi della sua assenza.

La riunione si considererà valida solo nel caso in cui siano presenti il Responsabile di Progetto e almeno il membro proponente e i sottoscritti alla riunione¹.

3.1.1 Casi particolari

- *Vicinanza² ad una milestone*

Le richieste di incontri che ricadono in questo periodo, possono non soddisfare i requisiti sopra scritti. In particolare basterà l'approvazione del Responsabile di progetto, e la riunione potrà svolgersi anche il giorno successivo dalla richiesta. Le comunicazioni di tali richieste dovranno seguire le modalità descritte nella sezione 2.1

- *Impossibilità di riunione per mancanza del numero legale*

In questo caso la riunione si considera annullata d'ufficio. Non ci sono modifiche alle regole sopra scritte o ulteriori provvedimenti

3.2 Incontri esterni

Come incontri esterni si intende un qualsiasi incontro fra un gruppo di rappresentanza del gruppo di progetto e i proponenti o il committente.

Sarà il Responsabile di Progetto a prendere accordi con il committente o i proponenti, utilizzando la procedura descritta nella sezione 2.2.

¹D'ora in poi questa condizione sarà definita "numero legale".

²Vicinanza: $\Delta t \leq 1$ settimana.

Ogni membro del gruppo può presentare richieste di incontro, motivando la necessità, al Responsabile di Progetto, il quale dovrà raccogliere almeno due conferme, e successivamente presenze, da parte dei membri del gruppo (escludendo il proponente dell'incontro) riguardo l'effettiva necessità dell'incontro. Nel caso in cui non si raggiungano le conferme necessarie la proposta sarà bocciata, altrimenti il Responsabile di Progetto contatterà la parte esterna di interesse per prendere accordi con la stessa. Le informazioni sull'incontro dovranno essere rese disponibili il prima possibile.

Sarà compito del Responsabile di Progetto redigere in seguito il verbale dell'incontro avvenuto.

4 Repository

Per lo svolgimento di qualsiasi progetto complesso è necessario l'utilizzo di un repository in cui conservare la documentazione e il codice tenendo traccia dello sviluppo.

1. Ubicazione:

L'indirizzo web in cui trovare il repository del progetto è: <https://github.com/ClockworkTeam/ClockWork>

2. Struttura:

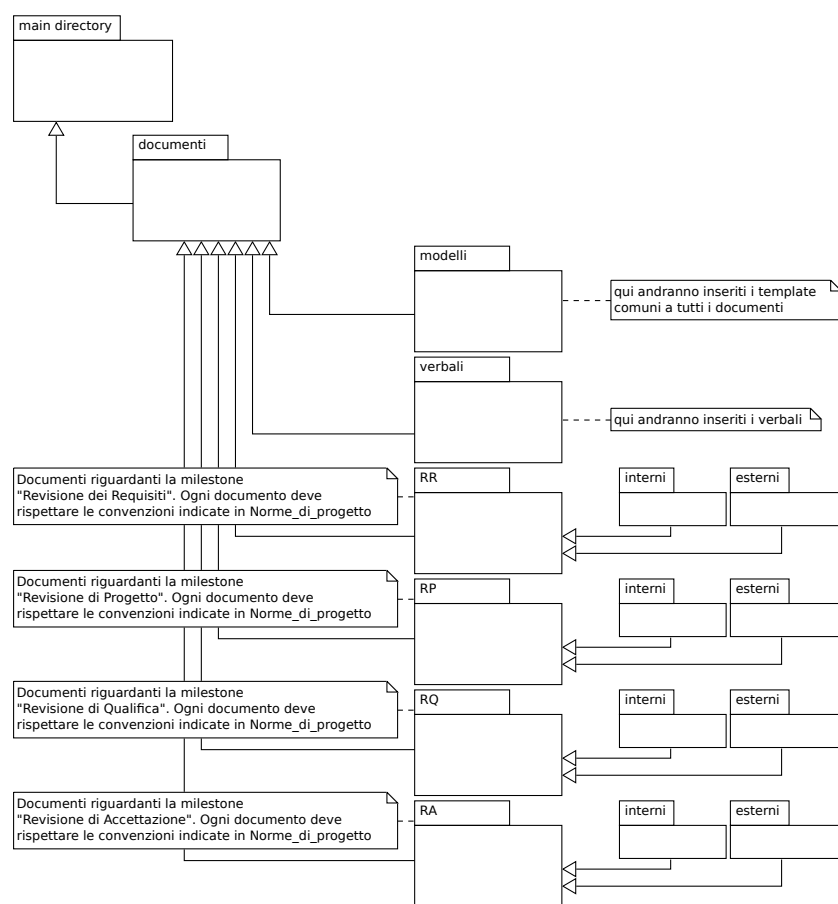


Figura 1: *Struttura del Repository*

Dalla cartella radice sarà possibile accedere alle cartelle Documenti e Codice.

4.1 Cartella documenti

La cartella documenti è così suddivisa:

- **Modelli:** in Modelli sono presenti file \LaTeX che definiscono la struttura comune a tutti i documenti e la cartella:
 - **img:** in Modelli/img sono presenti tutte le immagini da inserire nei documenti. (Se ci sarà una sezione apposta per le immagini fare riferimento ad essa)
- **Verbali:** in Verbali saranno presenti tutti i verbali degli incontri effettuati, suddivisi in Formali ed Informali. Questi saranno a loro volte suddivisi per mensilità in apposite cartelle
- **Cartelle Revisioni:** Ciascuna sottocartella di revisione (RR , RP , RQ ed RA) conterrà tutti i documenti utili alla revisione
Ogni sottocartella riguardante le revisioni conterrà le cartelle Interni ed Esterni:
 - **Interni:** si troveranno tutti i documenti formali necessari per l'organizzazione interna del gruppo ma non utili al proponente
 - **Esterni:** si troveranno tutti i documenti rivolti al proponente

4.2 Cartella Codice

La struttura che ordinerà la cartella codice sarà decisa a inizio della fase di progettazione.

5 Documenti

In questa sezione si presenteranno i vari standard adottati dal gruppo *Clockwork* per i vari documenti prodotti durante il processo di sviluppo del software.

5.1 Formattazione e copertina

Ogni documento dovrà essere redatto seguendo queste indicazioni³:

- **Frontpage:** ogni documento `LyX`, dovrà includere come primo oggetto il file `frontpage.lyx`⁴. Il file non dovrà essere spostato dalla sua posizione iniziale in quanto sarà `LyX`, assieme alle impostazioni contenute nel file `preface.tex`, a modificare automaticamente il contenuto del file. Nel file sono presenti le pagine di copertina e un'altra contenente le informazioni sul documento
- **Diario delle modifiche:** la struttura del diario delle modifiche è possibile trovarla nel file `diario_modifiche.lyx`. Questo file dovrà essere copiato ed incollato all'interno della cartella del proprio documento in lavorazione e incluso, all'interno del documento, subito dopo il frontpage. Inoltre dovrà essere modificato (la copia "locale") per mostrare correttamente il diario delle modifiche del documento
- **Impostazioni del "Preambolo di \LaTeX ":** le impostazioni che dovranno essere applicate ad ogni documento è possibile trovarle all'interno del file `preface.tex`. Questo file dovrà essere copiato-incollato all'interno della cartella del proprio documento in lavorazione. Inoltre dovrà essere modificato (la copia "locale") secondo quanto scritto al suo interno. All'interno di `LyX`, invece, si dovrà andare in Documento->Impostazioni..., selezionare nell'elenco di sinistra la voce Preambolo di \LaTeX , quindi scrivere `\include{preface}`
- **Loghi:** all'interno della cartella `img` si troveranno i file `logoBW.pdf` e `logo.pdf`. Si dovrà copiare la cartella all'interno della cartella del proprio documento

5.2 Norme tipografiche

Al fine di evitare incongruenze tra i vari file, si rimanda a questa sezione per le informazioni riguardanti l'ortografia, la tipografia e l'assunzione di uno stile uniforme in tutti i documenti.

³Ogni file citato in questa sezione è possibile trovarlo nel repository all'interno della cartella `/root/Documenti/Modelli`.

⁴Il percorso del file fornito a `LyX` non dovrà essere assoluto, ma relativo (`../../Modelli/frontpage.lyx`).

5.2.1 Stile del testo

- **Grassetto:** il grassetto è da utilizzare con parsimonia, prevalentemente per evidenziare passaggi estremamente importanti o da utilizzare come elemento immediatamente seguente agli elenchi per evidenziare l'oggetto trattato nel paragrafo
- **Corsivo:** il corsivo deve essere utilizzato per evidenziare passaggi o parole estremamente significativi a cui si vuole dare particolare enfasi e per riportare passaggi provenienti da fonti esterne
- **Sottolineato:** termini contenuti nel glossario, se lo stesso termine figura più volte nel documento viene sottolineato solo alla prima occorrenza⁵
- **Monospace:** il carattere `monospace` dovrà indicare un passaggio utile alla definizione di un formato, di uno standard adottato o per inserire del codice all'interno del documento
- **Maiuscolo:** una parola scritta maiuscola indicherà un acronimo. Non sono previsti ulteriori casi di parole scritte solo in maiuscolo⁶

5.2.2 Punteggiatura

- **Punteggiatura:** qualsiasi segno di punteggiatura non deve mai seguire un carattere di spaziatura, ma deve essere seguito da un carattere di spazio. Si ricorda che la lettera maiuscola segue esclusivamente il punto, il punto esclamativo ed il punto di domanda
- **Parentesi:** una qualsiasi frase racchiusa fra parentesi non deve iniziare con un carattere di spaziatura e non deve chiudersi con un carattere di punteggiatura e/o di spaziatura

5.2.3 Composizione del testo

- **Elenchi:** la prima parola che segue l'oggetto di indentazione deve avere la lettera maiuscola. La fine di ogni paragrafo deve essere priva di punteggiatura
- **Note a piè pagina:** ogni nota a piè di pagina dovrà cominciare con l'iniziale della prima lettera maiuscola, e non deve essere preceduta da alcun carattere di spaziatura. Ogni nota a piè pagina dovrà terminare con un punto

⁵Non si considerano valide come occorrenze le parole all'interno dei titoli delle sezioni, all'interno di percorsi.

⁶Fanno esclusione tutti i casi in cui la grammatica italiana preveda l'uso della maiuscola.

5.2.4 Formati ricorrenti

- **Path:** per gli indirizzi web completi andrà utilizzato il comando appositamente fornito da **LyX**, Inserisci->URL, mentre per eventuali indirizzi web relativi verrà utilizzata la formattazione monospace
- **Date:** se non specificato diversamente, dovrà essere espressa seguendo la “notazione italiana”, ovvero:

GG/MM/AAAA

dove:

- GG rappresenta il giorno trascritto utilizzando esattamente due cifre⁷
- MM rappresenta il mese trascritto utilizzando esattamente due cifre⁸
- AAAA rappresenta l'anno trascritto utilizzando esattamente quattro cifre

- **Nomi ricorrenti:**

- *Ruoli di progetto:* i vari ruoli di progetto andranno formattati utilizzando la prima lettera maiuscola per ogni parola che non sia una preposizione (es: Responsabile di Progetto)
- *Nomi dei documenti:* i vari nomi dei documenti andranno formattati utilizzando la prima lettera maiuscola per ogni parola che non sia una preposizione (es: Norme di Progetto)
- *Nomi dei file e delle cartelle:* nei nomi non andranno utilizzati caratteri speciali e/o caratteri accentati; qualora si dovesse utilizzare il nome di un file senza riferirlo con l'indirizzo completo si impone che venga formattato con un carattere monospace
- *Nomi propri:* l'utilizzo dei nomi propri dovrà seguire il formalismo “Cognome Nome”
- *Nome del gruppo:* ci si riferirà al gruppo *Clockwork* con la dicitura “gruppo” o “gruppo *Clockwork*”, con il nome del gruppo in corsivo
- *Nome del proponente:* ci si riferirà al proponente con “Zucchetti SPA” o con “proponente”
- *Nome del progetto:* ci si riferirà al progetto con **MyTalk** in grassetto

- **Sigle:** le sigle dei documenti dovranno essere utilizzate esclusivamente all'interno di diagrammi o tabelle (in modo da risparmiare spazio) secondo il seguente formalismo:

⁷Nel caso il numero potesse essere rappresentato con una singola cifra, si impone l'anteposizione di uno zero alla cifra.

⁸Vedi nota precedente.

- AR = Analisi dei Requisiti
- PdP = Piano di Progetto
- PdQ = Piano di Qualifica
- NdP = Norme di Progetto
- GL = Glossario

5.3 Componenti visive

5.3.1 Immagini

Tutte le immagini utilizzate all'interno dei documenti dovranno rigorosamente essere in formato PDF, PNG o EPS. Ogni immagine dovrà avere una breve didascalia in cui dovrà necessariamente comparire un nome identificativo per la tracciabilità di tale immagine nel testo. Si consiglia a tale scopo di utilizzare poche lettere e dei numeri incrementali.

5.4 Versionamento

Il formalismo utilizzato per esprimere l'avanzamento di versione dei vari documenti sarà il seguente:

$$v \{X\}.\{Y\}$$

dove:

- **X**: sta ad indicare il numero di uscite formali accumulate dal documento⁹
- **Y**: sta ad indicare i cambiamenti sostanziali¹⁰ susseguitisi dall'ultima uscita formale. Tale numero deve partire da zero e non ha un limite superiore

Per il corretto utilizzo del formalismo si dovranno seguire le seguenti normative:

- Nei nomi dei file approvati: `Nome_del_Documento_vX.Y.estensione`
- A piè pagina a seguire il nome dei documenti: Nome del Documento v X.Y
- All'interno di ogni documento in cui si nomina un altro documento, ma solo la prima volta in cui quel documento verrà riferito: Nome del Documento vX.Y

⁹Anche se il documento non ha subito sostanziali modifiche da un'uscita formale alla successiva, si impone che avanzi comunque di versione.

¹⁰Non si considera una modifica sostanziale un aggiornamento sulla punteggiatura o correzione ortografica.

5.5 Formattazione documenti

In ogni pagina di ogni documento formale, ad eccezione della copertina, dovranno comparire:

- **Logo** (versione in bianco e nero) come miniatura, da posizionare nel frontespizio sulla sinistra
- **Nome del progetto** da posizionare nel frontespizio sulla destra
- **Nome e versione¹¹ del documento** da posizionare a piè pagina sulla sinistra

Inoltre nelle pagine successive a quelle di apertura ¹² deve essere presente:

- **Numero di pagina** espresso nel formato:

n di tot

da posizionare a piè pagina sulla destra

Ogni documento prodotto deve contenere come pagine di apertura:

- **Copertina:**
 - *Nome del progetto* acquisito dal gruppo
 - *Logo del gruppo* sotto il quale deve comparire la mail per contattare il gruppo
 - *Nome e versione del documento*
- **Informazioni generali** del documento:
 - *Nome documento*
 - *Versione documento* ¹³
 - *Data creazione* ovvero il giorno in cui ha avuto inizio il lavoro sul documento
 - *Data ultima modifica* ovvero il giorno in cui è stata effettuata l'ultima modifica sul documento
 - *Uso documento* ovvero se il documento ha visibilità interna o esterna
 - *Redazione* ovvero quali membri del gruppo hanno contribuito alla stesura del documento
 - *Verifica* ovvero quali membri del gruppo hanno verificato il documento

¹¹Il formato utilizzato per tracciare le versioni è definito nella sezione 5.3.

¹²Le pagine di apertura sono quelle prima dell'indice, e l'indice stesso.

¹³Tale campo dovrà essere omissso nei documenti di tipo Verbale descritti nella sezione 5.6.1.

- *Approvazione* ovvero chi ha approvato la versione formale del documento
- *Lista distribuzione* ovvero la lista di persone alle quali dovrà pervenire il documento¹⁴

Inoltre in questa pagina è presente il *Sommario*, dove è presente una breve descrizione del documento.

- **Diario delle modifiche** ovvero una tabella che riporti le varie modifiche apportare al documento per giungere alla versione finale. Si tratta di una tabella ordinata in maniera decrescente: la prima riga dovrà contenere le informazioni riguardanti le ultime modifiche, mentre l'ultima riga dovrà contenere le informazioni riguardanti le prime modifiche effettuate al documento. Ogni riga dovrà contenere le seguenti informazioni:
 - *Autore*: dovrà contenere il nominativo di chi ha apportato la modifica in oggetto¹⁵
 - *Descrizione*: breve descrizione dei cambiamenti apportati
 - *Data*: quando la modifica è stata effettuata¹⁶
 - *Versione*: il numero della versione corrispondente al documento in cui è stata applicata la modifica
- **Indice dei contenuti**¹⁷

5.6 Tipi di documenti

5.6.1 Verbali incontri

Per Verbali degli Incontri si intendono quei documenti redatti dal Responsabile di Progetto in occasione di incontri esterni. Per tali documenti è prevista una sola stesura in quanto promemoria dell'incontro avvenuto e tematiche trattate: per tale motivo non è previsto versionamento. Nei verbali non sono presenti il diario delle modifiche e l'indice. Il documento dovrà essere denominato secondo il seguente criterio:

Verbale{data incontro¹⁸}

.

Le pagine copertina e informazioni del documento, di ogni verbale, devono sottostare alle regole descritte nella formattazione del documento¹⁹. A seguire dovrà essere presente una pagina con le informazioni logistiche dell'incontro, in particolare dovranno essere presenti:

¹⁴Per i documenti interni nella lista saranno presenti il gruppo e il committente, mentre per i documenti esterni sarà presente anche il proponente.

¹⁵Notazione definita nella sezione 5.2.4.

¹⁶Notazione definita nella sezione 5.2.4.

¹⁷Ad eccezione del glossario.

¹⁸In questo caso il formato della data dev'essere AAAAMMGG.

¹⁹Ad eccezione, ovviamente, del versionamento.

- **Data**²⁰

- **Luogo** espresso nel formato:

$$\{\text{città}\}(\{\text{provincia}\}) \{\text{indirizzo}\}\{\text{numero civico}\}$$

- **Ora ritrovo** espressa dal formalismo:

$$\{\text{hh}\}.\{\text{mm}\}$$
²¹

- **Durata incontro** espressa dal formalismo:

Durata dell'incontro: $\{x\}$ min

- **Partecipanti interni** indicante la lista dei membri del gruppo²² presenti all'incontro. La lista deve essere preceduta dall'etichetta:

Partecipanti del gruppo:

- **Partecipanti esterni** espressi dal formalismo:

Partecipanti esterni: $\{\text{Proponenti}\}$

- **oggetto dell'incontro** in cui si elencano i punti salienti dell'incontro con eventuali chiarimenti e/o decisioni prese a riguardo

5.6.2 Documenti informali

Tutti i documenti che non siano stati approvati dal Responsabile di Progetto dovranno essere considerati essere dei documenti informali. Questi documenti saranno considerati sempre ad uso interno. Saranno raccolti nel repository secondo la struttura definita alla sezione 4.

5.6.3 Documenti formali

Una volta approvati dal Responsabile di progetto i documenti si riterranno formali e pronti per essere distribuiti. Solo i documenti formali potranno essere distribuiti alla loro lista di distribuzione. Ogni volta che si appone una modifica ad un documento già approvato (quindi dichiarato formale), si dovrà inserire tale modifica nel diario delle modifiche, avvisando nel frattempo il Responsabile di Progetto che provvederà, nel caso non siano necessarie ulteriori modifiche, a riapprovare il documento e rilasciare una nuova versione formale di esso. Saranno raccolti nel repository secondo la struttura definita alla sezione 4.

²⁰Notazione definita nella sezione 5.2.4.

²¹I campi hh e mm dovranno assolutamente essere espressi tramite due cifre

²²Secondo il formalismo indicato nella sezione 5.2.4.

5.7 Procedura di formalizzazione dei documenti

Ogni volta che i redattori considerano pronto un documento, dovranno avvisare il Responsabile di Progetto, il quale assegnerà tale documento ai verificatori. Se il documento dovesse rispettare tutte le norme ivi descritte, il Responsabile di Progetto potrà approvare o meno il documento, oppure rimandarlo ai redattori per correzioni non normative.

5.8 Procedura di verifica dei documenti

PREMESSA: in questa sottosezione si elencherà solamente la procedura che verrà utilizzata per verificare i documenti. Per maggiori dettagli si faccia riferimento al [Piano_di_Qualifica_v1.0.pdf](#). I verificatori, una volta presa in consegna i documenti che hanno raggiunto una versione candidata al rilascio, dovranno applicare la seguente procedura:

- **Controllo ortografico:** il verificatore procederà al controllo ortografico del documento tramite Hunspell o Enchant, andando su Strumenti->Correttore Ortografico. Prima però dovrà controllare che la lingua del documento²³ sia correttamente impostata in italiano. Per fare ciò, dovrà andare su Documento->Impostazioni... e, nel menù laterale, selezionare la voce Lingua. Una volta selezionata la voce, dovrà guardare la lingua impostata e, nel caso, correggerla in “italiano”
- **Controllo lessicale:** il verificatore procederà ad un’attenta lettura dell’intero documento alla ricerca di errori lessicali
- **Diagrammi UML:** nel caso siano presenti dei diagrammi UML, il verificatore dovrà controllare che tali grafici rispettino le norme descritte in sezione 12.2.4 procedendo tramite attenta osservazione di essi

5.9 Incompatibilità

Durante le varie fasi del progetto potrebbero emergere problematiche riguardanti le incompatibilità fra ruoli, ovvero redattori che verificano i documenti scritti da loro stessi piuttosto che documenti approvati da chi ha verificato o contribuito alla redazione dei documenti stessi. Per questo motivo vengono poste le seguenti norme che serviranno a specificare sia le norme per l’assegnazione del personale di redazione, verifica e approvazione di un documento, sia le procedure da adottare nel caso ci siano conflitti e/o incompatibilità.

- **Redattori-verificatori:** i redattori di una determinata versione di un documento non possono esserne anche i verificatori della stessa versione²⁴, tranne nei seguenti casi:

²³Poiché i documenti saranno formati da un file per ogni capitolo, il verificatore dovrà effettuare questa procedura per ogni singolo file.

²⁴Potranno tuttavia verificare versioni future del documento.

- **Mancanza di verificatori:** nel caso in cui non sia possibile assicurare che i documenti vengano verificati da persone che non abbiano contribuito alla redazione del documento, il Responsabile di Progetto procederà alla nomina di almeno un verificatore controllando il Diario delle modifiche del suddetto documento e scegliendo il primo che ha smesso di contribuire al documento; costui sarà il verificatore. Nel caso in cui ci sia bisogno di più di un verificatore, si applicherà la stessa norma, selezionando la seconda persona che ha smesso di contribuire al documento e così per le successive nomine

5.10 Informazioni aggiuntive sui documenti

In questa sezione verranno aggiunte delle informazioni aggiuntive sulla redazione dei documenti e/o su alcune componenti dei documenti.

- Sulle informazioni generali dei documenti: come scritto precedentemente all'interno delle informazioni generali di un documento dovranno essere inserite la lista dei redattori, dei verificatori e degli approvatori del documento. Queste liste non dovranno essere cancellate da una versione all'altra, ma dovranno essere integrate con i nomi delle persone coinvolte nell'aggiornamento di versione. Potrebbe succedere di conseguenza che persone coinvolte nella redazione di un documento compaiano anche fra i verificatori²⁵

²⁵Vedere sezione precedente.

6 Analisi dei Requisiti

L'analisi dei Requisiti sarà redatta dagli analisti.

In tale documento dovranno comparire ordinatamente tutti i requisiti rilevati dal Capitolato e/o dagli incontri effettuati col proponente.

Ogni requisito dovrà essere il più completo e chiaro possibile. Per tale motivo si richiede che per ogni requisito siano specificati:

- **Fonti:** ovvero da che luogo è emerso il requisito
- **Classificazione:** ovvero raggruppare i requisiti in insiemi coerenti in modo da poter definire una segnatura specifica, univoca e gerarchica secondo il seguente formalismo:

$R\{AMBITO\}\{TIPO_REQUISITO\}\{PRIORITÀ\} \{GERARCHIA\}$

dove:

- AMBITO: distinguerà l'ambito per cui sarà definito il requisito:
 - * **U** indicherà qualsiasi tipo di utente
 - * **G** indicherà un requisito generale di sistema
 - TIPO_REQUISITO: distinguerà la tipologia del requisito:
 - * **F** indica un requisito funzionale
 - * **Q** indica un requisito di qualità
 - * **V** indica un requisito di vincolo
 - PRIORITÀ: distinguerà la priorità associata al requisito:
 - * **O** indicherà un requisito obbligatorio, ossia un requisito la cui implementazione è fondamentale per la riuscita dell'applicazione
 - * **D** indicherà un requisito desiderabile, ossia un requisito la cui implementazione è gradita ma non fondamentale e la cui realizzazione è da intendersi di importanza secondaria rispetto ai requisiti facoltativi
 - * **F** indicherà un requisito facoltativo, ossia un requisito la cui implementazione non è da ritenersi vincolante per la riuscita del progetto: saranno requisiti ricavati dal capitolato o dal proponente e che il gruppo si impegnerà ad implementare qualora le risorse disponibili saranno sufficienti
 - GERARCHIA: distinguerà la gerarchia presente tra i vari requisiti
- **Descrizione grafica:** utilizzare il formalismo di UML per creare diagrammi di casi d'uso che facilitino la comprensione di tali requisiti. I diagrammi dei casi d'uso utilizzeranno una nomenclatura simile alla precedente:

$UC\{GERARCHIA\}$

dove:

- GERARCHIA: distinguerà la gerarchia presente tra i vari casi d'uso
- **Descrizione didascalica:** si dovrà accompagnare ogni diagramma di caso d'uso con una descrizione dove dovranno comparire le seguenti informazioni:
 - Attori coinvolti
 - Scopo e descrizione del requisito
 - Precondizione
 - Postcondizione
 - Flusso principale degli eventi
 - Scenari alternativi

7 Studio di fattibilità

In questo documento si richiede lo studio di ogni capitolato analizzando i seguenti punti:

- **Fattibilità tecnica:** comprende la valutazione dei rischi legati alle tecnologie richieste per soddisfare i requisiti e alla difficoltà stimata da parte del gruppo di apprendere l'utilizzo delle stesse
- **Fattibilità socioeconomica:** comprende la valutazione dei rischi legati agli aspetti socioeconomici del progetto come, ad esempio, i costi da affrontare e la concorrenza di altri fornitori
- **Fattibilità implementativa:** comprende la valutazione dei rischi legati alla soddisfaccibilità dei requisiti del capitolato e quindi la stima della capacità del gruppo di completare le richieste obbligatorie, desiderabili e opzionali
- **Altro:** qualsiasi altro fattore di rischio stimato dagli analisti non compreso nei casi precedenti

7.1 Rischi

Per ogni fattore enunciato precedentemente dovranno essere catalogati i rischi come viene descritto nel seguente protocollo:

- **Nome del rischio:** sintetizza il rischio individuato con un nome significativo
- **Descrizione del rischio:** descrive brevemente il rischio individuato
- **Stima di probabilità:** l'analista indica la probabilità che si verifichi la situazione di rischio in esame con un parametro compreso nell'insieme { *Certa*, *Alta*, *Media*, *Bassa*, *Nulla* }
- **Stima di incidenza:** l'analista indica quanto ritiene grave il verificarsi della situazione di rischio in esame con un parametro compreso nell'insieme { *Catastrofica*, *Alta*, *Media*, *Bassa*, *Trascurabile* }
- **Giustificazione della stima:** l'analista spiega le motivazioni dell'assegnamento alle stime di probabilità e di incidenza assegnate
- **Analista:** nome dell'analista che sarà contattato dal Responsabile di Progetto (o dal Verificatore per conto del Responsabile di Progetto) nel caso in cui le valutazioni appaiano inesatte o poco giustificate
- **Riferimenti**²⁶: elenco puntato di tutte le fonti utilizzate, per consentire una futura verifica delle fonti

²⁶Questo punto è facoltativo.

7.2 Vantaggi

Per ogni fattore enunciato precedentemente dovranno essere catalogati i vantaggi come viene descritto nel seguente protocollo:

- **Nome del vantaggio:** sintetizza il vantaggio individuato con un nome significativo
- **Descrizione del vantaggio:** descrive brevemente il vantaggio individuato
- **Stima di incidenza:** indica quanto l'analista ritiene incidente nella valutazione del vantaggio complessivo il vantaggio in esame con un parametro compreso nell'insieme {*Ottimale, Alta, Media, Bassa, Trascurabile*}
- **Giustificazione della stima:** l'analista spiega le motivazioni dell'assegnamento alla stima di incidenza assegnata
- **Analista:** nome dell'analista che sarà contattato dal Responsabile di Progetto (o dal Verificatore per conto del Responsabile) nel caso in cui le valutazioni appaiano inesatte o poco giustificate
- **Riferimenti**²⁷: elenco puntato di tutte le fonti utilizzate, per consentire una futura verifica delle fonti

Una volta terminato lo studio di fattibilità di tutti i capitolati, l'analista redattore del documento finale dovrà inserire una conclusione in cui si esporrà:

- una **valutazione complessiva** dei rischi e dei vantaggi di ciascun capitolato
- il **capitolato scelto**
- un'**analisi sintetica** di vantaggi e rischi di ciascun capitolato

²⁷Questo punto è facoltativo.

8 Progettazione

Durante la fase di progettazione i progettisti dovranno adeguarsi alle seguenti specifiche:

- **Diagrammi:** si andrà ad utilizzare il linguaggio UML per definire:
 - *Diagrammi delle classi:* dovrà essere presente all'interno dei documenti l'intera architettura generale e di dettaglio
 - *Diagrammi di flusso:* presente nel caso in cui l'azione di codifica dei programmatori dovesse portare aleatorietà e conseguentemente non garantire il corretto funzionamento dell'applicazione supposta dall'architettura
 - *Diagrammi di package:* presenti sia nell'architettura generale che di dettaglio. I vari package saranno definiti in maniera univoca per poterli distribuire a vari codificatori durante la fase di codifica differenti package che interagiscono tra loro
- **Design di pattern:** per i vari design pattern andremo a specificare:
 - una *descrizione generale* per presentare brevemente la struttura del design di pattern
 - una *motivazione* di tale design di pattern
 - il *contesto applicativo* dove andremo ad utilizzarlo
- **Classi di verifica:** da sviluppare quando possibile, soprattutto per le classi generali, delle classi fittizie da utilizzare durante la fase di verifica e come prototipo
- **Stile di progettazione:** per la semplificazione degli schemi e per la prossima fase di progettazione si cercheranno di rispettare le seguenti norme
 - *Ricorsione:* si cercherà di evitare l'utilizzo della ricorsione qualora fosse possibile una soluzione: nel caso si andasse ad utilizzare la tecnica della ricorsione si dovrà stimare l'utilizzo di memoria che si andrà ad occupare ed in caso la memoria occupata fosse eccessiva si andrà ad eliminare la ricorsione
 - *Concorrenza:* si andranno a fornire i diagrammi di flusso ed una stima delle risorse necessarie. Nel caso in cui i benefici ricavati dalla concorrenza non siano equivalenti o superiori alle risorse utilizzate si andrà ad eliminare la concorrenza
 - *Annidamento di chiamata:* non dovrà esserci una profondità di annidamento maggiore di dieci (10)

- *Flussi di condizione:* per la chiarezza e la semplicità di verifica del codice, qualora vengano utilizzati costrutti condizionali (if-then-else) non ci dovrà essere un annidamento maggiore di cinque (5)
- *Numero di parametri:* non dovranno essere presenti metodi con un numero di parametri maggiore di dieci (10)

9 Codifica

9.1 Intestazione del file

Ogni file di codice corrisponderà esattamente ad ogni singola classe, ed inizierà con un'intestazione che dovrà rispecchiare il seguente standard:

```
/*
 * Nome: {nome del file}
 * Package: {package di appartenenza}
 * Autore: {autore del file}
 * Data: {data di creazione del file}
 * Versione: {versione del file}
 *
 * Modifiche:
 *


| Data   | Programmatore | Modifiche         |
|--------|---------------|-------------------|
| GGMMAA | NomeCognome   | - [label] metodo1 |
|        |               | - [label] metodo2 |
|        |               | - ....            |


 *
 */
```

dove:

- **Nome:** sarà il nome del file comprensivo di estensione
- **Package:** sarà comprensivo della gerarchia del package
- **Autore:** sarà l'autore del file e non necessariamente il programmatore che sta modificando il file attualmente
- **Data:** sarà la data di creazione del file
- **Versione:** indica la versione attuale del file
- **Modifiche:** rappresenta la tabella di avanzamento del file. Per convenzione:
 - **Data:** rappresenta la data dell'avvenuta modifica nel formato GGMMAA con le cifre:
 - * GG che rappresentano il giorno con eventuali zeri iniziali
 - * MM che rappresentano il mese con eventuali zeri iniziali
 - * AA che rappresentano l'anno

- **NomeCognome**: rappresenta il programmatore che ha effettuato le modifiche. Conterrà la prima lettera del nome seguita dalle prime quattro lettere del cognome²⁸
- **Modifiche**: rappresenta la lista dei cambiamenti. Per ogni riga dovrà esserci un solo cambiamento. *Label* potrà essere:
 - /+/:* per indicare la creazione del metodo
 - /-/:* per indicare l'eliminazione del metodo
 - /*/:* per indicare la modifica del metodo

9.2 Convenzioni di codifica

Le convenzioni di codifica saranno formalizzate all'inizio della fase di progettazione.

²⁸Nel caso in cui il cognome sia più corto di quattro lettere non verranno inseriti altri caratteri.

10 Glossario

Il glossario andrà formattato come deciso nella sezione 5 di questo stesso documento. Il glossario sarà unico e riassuntivo per tutti i documenti. Il glossario riporterà al suo interno tutte le definizioni, in ordine lessicografico, delle parole che possono generare confusione o ambiguità all'interno dei documenti, che appariranno sottolineate alla loro prima occorrenza nel documento.

10.1 Inserimento vocaboli

Data l'universalità del glossario rispetto a tutti i documenti redatti dal gruppo *Clockwork*, viene nominato Responsabile del Glossario il Responsabile di Progetto. Per poter chiedere l'inserimento di un nuovo vocabolo si dovrà quindi fare richiesta al Responsabile di Progetto che si appresterà a decidere se la parola presa in esame possa essere fonte di ambiguità e/o confusione. Per procedere all'inserimento di una nuova parola il Responsabile di Progetto dovrà seguire le seguenti regole:

- Verificare la presenza della parola presa in esame
- Inserire il vocabolo nella posizione che gli compete

11 Gestione delle attività

Le attività saranno gestite tramite il servizio di ticketing offerto da GitHub²⁹ in quanto, assieme al servizio di milestone si può tenere facilmente traccia dello stato di avanzamento dei lavori.

Questi strumenti saranno utilizzati come dice il seguente protocollo:

1. **Creazione milestone:** sarà il Responsabile di Progetto a creare una *milestone* per la successiva revisione a cui il gruppo *Clockwork* ha intenzione di partecipare. Lo stato di avanzamento si potrà vedere dal numero di ticket completati rispetto al numero di ticket aperti.

Per fare ciò, bisogna accedere al repository, andare su Issues -> Milestones -> Create a new milestone.

La *milestone* dovrà contenere le seguenti informazioni:

- Nome della milestone
- Nome del Responsabile
- Data di conclusione della milestone

2. **Creazione di un ticket:** il Responsabile dovrà creare un ticket per ogni compito che dovrà essere assegnato ad un membro del gruppo *Clockwork*. Le segnalazioni di *Anomalie* e *Discrepanze* sono altri casi previsti per la creazione di un ticket, e queste possono essere fatte da un qualsiasi membro del gruppo.

Per creare un ticket bisogna accedere al repository, andare su Issues -> New Issue.

La creazione di un ticket va fatta definendo obbligatoriamente i campi:

- **Title:** nome del compito assegnato
- **Assignee:** colui a che dovrà concludere il ticket
- **Milestone:** la *milestone* per la quale il compito dovrà essere terminato
- **Label:** impostate a seconda di precisi argomenti nella fase di progettazione, alla creazione del ticket andrà scelta quella più appropriata
- **Text:** dovrà contenere le seguenti informazioni:
 - Descrizione del compito affidato
 - Data di inizio prevista
 - Data di fine prevista
 - Collega³⁰

²⁹Questo servizio sarà usato anche per le comunicazioni fra membri singoli come descritto nella sezione Comunicazioni interne.

³⁰Questo campo solo se più di un membro è affidato allo stesso compito, dunque è facoltativo.

3. **Esecuzione compito e risoluzione di un ticket:** ogni membro del gruppo dovrà visionare i ticket a lui assegnati e inserire un nuovo commento per ogni aggiornamento sullo stato del ticket. Ogni volta che viene concluso il compito, si dovrà aggiungere l'apposita label “*Completato*” e menzionare il ticket nel relativo commit. Sarà il Responsabile di Progetto che dovrà confermare l'effettiva chiusura del ticket solo se il compito è stato eseguito. La label aiuta il Responsabile di Progetto ad individuare i ticket da chiudere
4. **Ticket di verifica:** ticket di questa tipologia verranno creati secondo quanto scritto precedentemente
5. **Chiusura milestone:** una milestone verrà considerata terminata una volta che tutti i ticket creati sono stati chiusi. Quando una milestone viene conclusa, il Responsabile di Progetto dovrà ripartire dal punto numero uno di questa procedura

12 Ambiente di progetto

Nella seguente sezione verrà descritto in dettaglio l'ambiente di sviluppo che il gruppo *Clockwork* andrà ad utilizzare per lo svolgimento del progetto

12.1 Ambiente generale

Questa sezione è dedicata a caratteristiche dell'ambiente di progetto che si ripercuotono su tutti gli altri ambienti

12.1.1 Sistema operativo

Il sistema operativo utilizzato durante lo sviluppo del progetto non è vincolante, i vari membri del gruppo avranno a disposizione libertà di scelta sulla piattaforma di sviluppo. Inoltre l'utilizzo di SO diversi permetterà di verificare la portabilità su più SO.

12.2 Ambientale documentale

12.2.1 Scrittura documenti

La scrittura dei documenti avverrà tramite codice L^AT_EX utilizzando l'editor L^YX (≥ 2.0), (<http://www.lyx.org>). Verrà creato un documento per ogni capitolo di ogni documento, successivamente verranno uniti sempre tramite L^YX generando un documento in formato PDF.

12.2.2 Verifica ortografica

Per la verifica ortografica verranno utilizzati Hunspell ($\geq 1.3.2$ -4build1) (<http://hunspell.sourceforge.net/>) ed Enchant ($\geq 1.6.0$ -7build1) (<http://abisource.com/projects/enchant/>) all'interno dell'editor L^YX lanciandolo attraverso il menu Strumenti->Correttore Ortografico o premendo il tasto F7.

12.2.3 Pianificazione

Per quanto riguarda la pianificazione delle attività legate allo sviluppo del progetto e la gestione delle risorse verrà utilizzato lo strumento Microsoft Project 2010, strumento fornito dal servizio MSDNAA di Microsoft in collaborazione con il Dipartimento di Matematica Pura ed Applicata dell'Università degli studi di Padova. Consci del fatto che il programma è compatibile solo con sistemi operativi Windows in caso di necessità sarà possibile utilizzare una macchina virtuale Windows XP (fornito anch'essa dalla MSDNAA) con installato il programma.

12.2.4 Grafici UML

I grafici UML verranno realizzati tramite programma ArgoUml (≥ 0.34), scelto in quanto software open source e multiplatforma e perché in grado di esportare

i grafici in formato vettoriale. Tuttavia il programma non supporta ufficialmente lo standard UML2, quindi verrà prestata attenzione nell'espressione delle notazioni grafiche affinché rispettino le specifiche 2.0.

12.2.5 Mockup

Per la realizzazione di mockup dell'interfaccia grafica verrà utilizzato il programma open source Pencil ($\geq 2.0.3$).

12.2.6 Documentazione semi-automatica

Lo strumento di documentazione semi-automatica impiegato sarà Javadoc (≥ 1.5), funzionalità di Java che permette di creare tutta la documentazione relativa ad una porzione di codice a partire dai commenti.

12.3 Ambiente di sviluppo

12.3.1 Strumenti di versionamento

Come strumento di versionamento si è deciso di utilizzare git in quanto veloce da utilizzare e di facile apprendimento. Per lo sviluppo collaborativo abbiamo deciso di appoggiarci al servizio Github (<https://github.com/>) che fornisce non solo un repository *git*, ma anche strumenti utili alla collaborazione fra più persone, come il servizio di *ticket*, *wiki* e *milestone*. Per quanto riguarda l'uso di *git* e *Github* sui computer di sviluppo, si è deciso l'uso della versione ufficiale rilasciata dal team di sviluppo di *git* ($\geq 1.7.8$) e le interfacce grafiche per i rispettivi sistemi operativi: *Github for Windows* (≥ 1.0), *Github for Mac* (\geq The Snappy, Actually) e *Git-Cola* ($\geq 1.4.3.5 - 1$) (Linux).

12.3.2 Ambiente di codifica

Per la scrittura del codice HTML si è deciso di utilizzare Geany (≥ 0.21), mentre per la scrittura del codice Java della parte lato server si è deciso di utilizzare l'IDE Eclipse ($\geq 3.8.0$).

12.4 Ambiente di verifica e validazione

Di seguito vengono elencati gli strumenti scelti per la verifica e la validazione

12.4.1 Analisi statica

L'analisi statica verrà eseguita tramite il programma Yasca (≥ 2.1). (<http://www.scovetta.com/yasca.html>) programma open source che permette l'analisi statica del codice HTML.

12.4.2 Test

Per i test su browser useremo il plugin per *Google Chrome* Chromedriver (\geq 23.0.1240.0) (<http://code.google.com/p/chromedriver>).

12.4.3 Analisi dinamica

L'analisi dinamica verrà effettuata con il plugin per *Google Chrome* Firebug Lite (\geq 1.4) (<http://getfirebug.com/releases/lite/chrome>).

12.4.4 Validazione

Per la verifica del codice HTML5 e CSS versione 3 utilizzeremo i validatori del W3C raggiungibili ai seguenti indirizzi: <http://validator.w3.org/>, <http://jigsaw.w3.org/css-validator/>.