

3 Informações para Coordenação da Execução de Testes

Diversas ferramentas oferecidas na literatura têm auxiliado na coordenação da execução dos testes de software, como, por exemplo, ferramentas de integração contínua (Duvall et al., 2007) e de gerenciamento de teste (Black, 2009). No entanto, para que esse controle seja realizado, geralmente um conjunto de informações é utilizado, como, por exemplo, (i) o momento que a execução dos testes será iniciada, (ii) quais testes serão executados, (iii) a ordem de execução dos testes a ser respeitada, (iv) assim como o log e o formato como os resultados dos testes serão armazenados. Apesar dessas ferramentas apresentarem bastante utilidade em projetos de teste, elas não ajudam na coordenação dos testes quando SUTs precisam ter autonomia para se autotestar, assim como mencionado por (Denaro et al. 2007; Wen et al. 2005). Portanto, para que SUTs adquiram autonomia e pró-atividade para controlar a execução dos seus testes, o paradigma de agentes de software (ver Seção 2.5) torna-se extremamente útil.

Dessa forma, para que agentes de software não tenham dificuldades em coordenar a execução dos testes, informações úteis sobre os testes disponíveis devem ser usadas. Na Seção 3.1 são apresentados os procedimentos adotados para identificar essas informações. Enquanto que na Seção 3.2 são apresentadas em detalhe as informações identificadas.

3.1. Procedimentos Empíricos

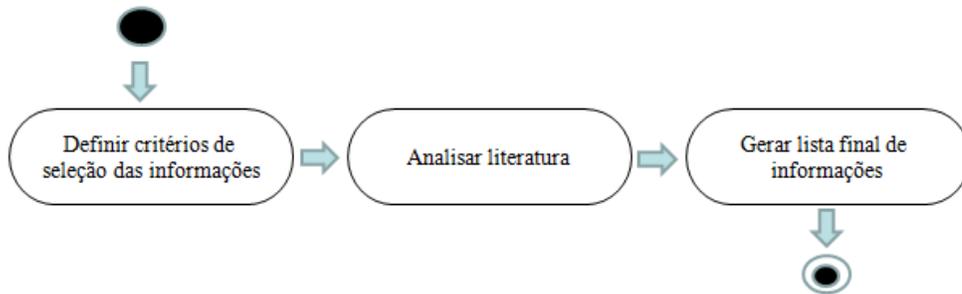


Figura 4. Procedimentos empíricos para identificar informações para coordenação de teste a partir da execução de agentes de software.

A fim de permitir a identificação de informações úteis para a coordenação de teste realizada por agentes de software, um procedimento dividido em três etapas e que está ilustrado na Figura 4 foi adotado. A primeira etapa definiu critérios que permitissem considerar informações como úteis para a coordenação dos testes. Após reuniões envolvendo dois especialistas da área de sistemas multiagentes e um na área de teste de software, dois dos critérios listados a seguir deveriam ser atendidos pelas informações a serem identificadas na literatura. Os critérios definidos foram os seguintes.

- Estar presente em dois ou mais documentos analisados sobre gerência de testes de software.
- Ser utilizada em duas ou mais ferramentas de gerenciamento de teste ou de integração contínua.
- Permitir que agentes executem testes de software.
- Ajudar agentes de software a decidir quais testes deverão ser executados para validar algum artefato do SUT.
- Permitir que agentes analisem os resultados dos testes executados para tomadas de decisão.

Após a definição dos critérios, a etapa 2 envolveu a análise de conhecidas ferramentas de integração contínua (Hudson, Continuum, Cruise Control e RTC) (Costa et al., 2008a; RTC, 2012) e de gerenciamento de teste (Rational TestManager e Rational Quality Manager) (RTM e RMT, 2012; RQM, 2012), os modelos de maturidade TMM (Burnstein et al., 2012; Burnstein e Miller, 2012) e

TMMi (TMMi, 2012), assim como referências adicionais da área de testes de software, como, (829-2008, 2012; Graham et al., 2008; Black, 2008; Black, 2002). A partir dessas análises, informações foram identificadas baseadas nos critérios definidos na etapa 1.

Já na etapa 3, última etapa do procedimento ilustrado na Figura 4, reuniões envolvendo os responsáveis por definir os critérios (etapa 1) foram realizadas para definir a lista final das informações a serem consideradas. Na seção a seguir tais informações são apresentadas em detalhe, enquanto que no capítulo 5 os relacionamentos entre elas são explorados em detalhe, já que um novo modelo conceitual baseado nesse levantamento é apresentado.

3.2. Informações Identificadas

A seguir são apresentadas as informações identificadas a partir do procedimento apresentado na seção anterior.

- **Testes a serem executados:** Para que seja possível coordenar a execução dos testes, o agente de software deve conhecer quais testes poderá executar. Assim, para que não haja dificuldade na identificação dos testes, cada um deles deve ter um identificador único (TMMi, 2012).
- **Ferramentas necessárias para executar testes:** Como testes automatizados podem ser executados a partir de diferentes ferramentas, o agente coordenador precisa conhecê-las para que seja capaz de executar e realizar os tratamentos necessários desses testes.
- **Critério de seleção dos testes para execução:** Diversos critérios podem ser adotados para orientar a decisão do agente sobre quais testes deverão ser executados em um dado momento. A seguir são apresentados exemplos de critérios: (i) somente testes de regressão poderão executados, (ii) somente testes com prioridade alta e que sejam de regressão, ou (iii) somente aqueles com prioridade alta, unitários e que sejam testes de segurança.

- **Níveis de teste:** Cada nível de teste (unitário, integração, sistema e aceitação) é um conjunto de atividades de teste organizadas e gerenciadas conjuntamente. Portanto, conhecer quais testes estão relacionados a cada nível torna-se importante, já que critérios de execução podem usar como base essa informação para verificar se testes podem ou não serem executados.
- **Tipos de Teste:** Os tipos de teste são grupos de atividades que testam um componente ou sistema focando em um objetivo de teste específico, como, por exemplo, testes funcionais, usabilidade, regressão, etc. Assim, tipos de teste também podem ser usados como base pelos agentes para decidir quais testes executar.
- **Testes manuais e automatizados:** Testes podem ser executados manualmente ou de forma automática. Testes manuais são aqueles que precisam de uma pessoa para executar um conjunto de passos para informar ao final o sucesso ou falha no teste. Já testes automatizados são aqueles executados a partir de uma ferramenta e que não precisam de uma interferência humana. Portanto, quando um agente identifica se um teste é manual ou automatizado, ele sabe se há ou não a necessidade da participação de uma pessoa. Além disso, essa informação pode ser utilizada em critérios de execução.
- **Prioridade de execução:** Para que agentes de software possam conhecer a ordem de execução dos testes a partir de algum critério definido por um testador ou desenvolvedor, definir a prioridade de execução de cada teste torna-se importante. Diferentes prioridades podem ser definidas, como, por exemplo, alta, média, baixa e muita baixa. Geralmente sistemas de gerenciamento de issues, como Atlassian JIRA (AJ, 2012) e Rational Quality Manager (RQM, 2012) usam prioridades para ajudar na coordenação e execução de tarefas.
- **Artefatos a serem testados:** Outra informação importante para agentes coordenadores é conhecer a rastreabilidade de quais artefatos (ex: web-service, classe, método, etc.) do SUT serão testados por quais testes. A partir dessa rastreabilidade, o agente pode controlar quais testes serão necessários para validar algum artefato específico.

- **Dependências entre testes:** Como dependências de execução podem existir entre testes de software, assim como mencionado pelo TMMi e IEEE 829-2008, o agente deve conhecer tais dependências para definir os adequados fluxos de execução (suites). Dessa forma, problemas em execuções dos testes podem ser evitados.
- **Contexto de execução:** Para que um agente coordene a execução de testes, torna-se necessário conhecer o foco desses testes, isto é, o sistema/componente em teste, assim como a versão (release/patch) considerada do produto (829-2008, 2012).
- **Dados de entrada e saída usados pelos testes:** Cada teste pode ser composto por diferentes dados de entrada que resultam em dados de saídas, caracterizando distintas condições de teste. Assim, conhecer quais dados utilizar e onde tais dados estão disponíveis para uso (ex: banco de dados ou arquivo), ajuda ao agente coordenar a execução dos testes. Dessa forma, dois conjuntos de dados podem ser usados por um mesmo teste, caracterizando situações distintas de validação em um mesmo SUT.
- **Risco relacionado à execução do teste:** Risco está diretamente ligado ao artefato em teste e que pode resultar em consequências futuras negativas. Assim, conhecer quais testes poderão indicar maiores riscos ao SUT quando possíveis falhas forem identificadas, permite ao agente usar essa informação em algum critério de execução para escolher quais testes executar.
- **Logs com os resultados dos testes executados:** Quando testes são executados para validar um ou mais artefatos, logs responsáveis por descrever os resultados desses testes devem estar disponíveis para que decisões possam ser tomadas em tempo de execução pelo SUT. A partir desses resultados, agentes podem decidir se há a necessidade de executar mais testes para tomar alguma decisão.

3.3. Considerações Finais

Neste capítulo foram apresentadas informações consideradas úteis para a coordenação de testes a partir da execução de agentes de software. A fim de identificá-las, o procedimento descrito na Seção 3.2 foi adotado, enquanto que na Seção 3.3, tais informações foram apresentadas em detalhe.

No próximo capítulo é apresentado o Java self-Adaptive Agent Framework for self-Test (JAAF+T). JAAF+T é um framework que permite a criação de agentes de software autoadaptáveis capazes de coordenar a execução de testes responsáveis por validar suas adaptações a partir da manipulação de informações apresentadas na Seção 3.2. Os detalhes de como essas informações são utilizadas pelo framework para permitir a coordenação dos testes são apresentados a seguir.