UNIVERSIDADE ESTADUAL PAULISTA
INSTITUTO DE BIOCIÊNCIAS, LETRAS E CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO E ESTATÍSTICA

Prototipação de Software

Engenharia de Software

2o. Semestre de 2005

Prototipação de Software

- Desenvolvimento rápido de software para validar os requisitos.

Objetivos

- Compreender o papel da prototipação em diferentes tipos de projetos de desenvolvimento.
- Discutir a prototipação evolucionária e a prototipação descartável.
- Introduzir três diferentes técnicas de prototipação.
- Explicar a técnica de prototipação no desenvolvimento de interface com o usuário

Tópicos abordados

- Prototipação no processo de software
- Técnicas de prototipação rápida
- Prototipação de interface com o usuário

Prototipação de sistema

- Prototipação é o desenvolvimento rápido de um sistema.
- No passado, protótipo tinha a finalidade exclusiva de avaliar os requisitos, assim o desenvolvimento tradicional era necessário.
- Atualmente , os limites entre a prototipação e o desenvolvimento normal do sistema, muitas vezes, são indefinidos e muitos sistemas são desenvolvidos usando uma abordagem evolucionária.

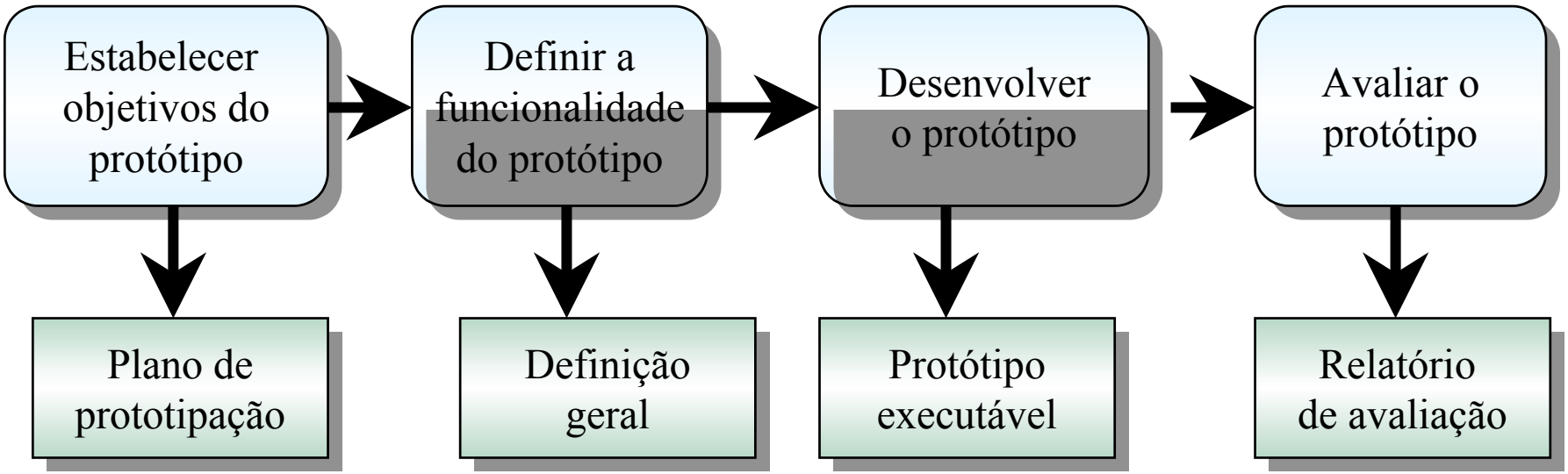
Usos de protótipos de sistemas

- O principal uso é ajudar os clientes e desenvolvedores entender os requisitos para o sistema.
  - Levantamento de requisitos. Usuários podem experimentar o protótipo para ver como o sistema pode apoiar o seu trabalho
  - Validação de requisitos. O protótipo pode revelar erros e omissões nos requisitos.
- A prototipação pode ser considerada como uma atividade de redução de riscos que reduz os riscos nos requisitos.

Benefícios da prototipação

- Equívocos entre os usuários de software e desenvolvedores são expostos.
- Serviços esquecidos podem ser detectados e serviços confusos podem ser identificados.
- Um sistema funcionando está disponível nos primeiros estágios no processo de desenvolvimento.
- O protótipo pode servir como uma base para derivar uma especificação do sistema com qualidade de produção.
- O protótipo pode ser usado para treinamento do usuário e teste de sistema.

Processo de desenvolvimento de protótipo



Benefícios da prototipação

- Melhoria na facilidade de uso do sistema;
- Maior aproximação do sistema com as necessidades dos usuários;
- Melhoria da qualidade do projeto;
- Melhoria na facilidade de manutenção, e
- Redução no esforço de desenvolvimento

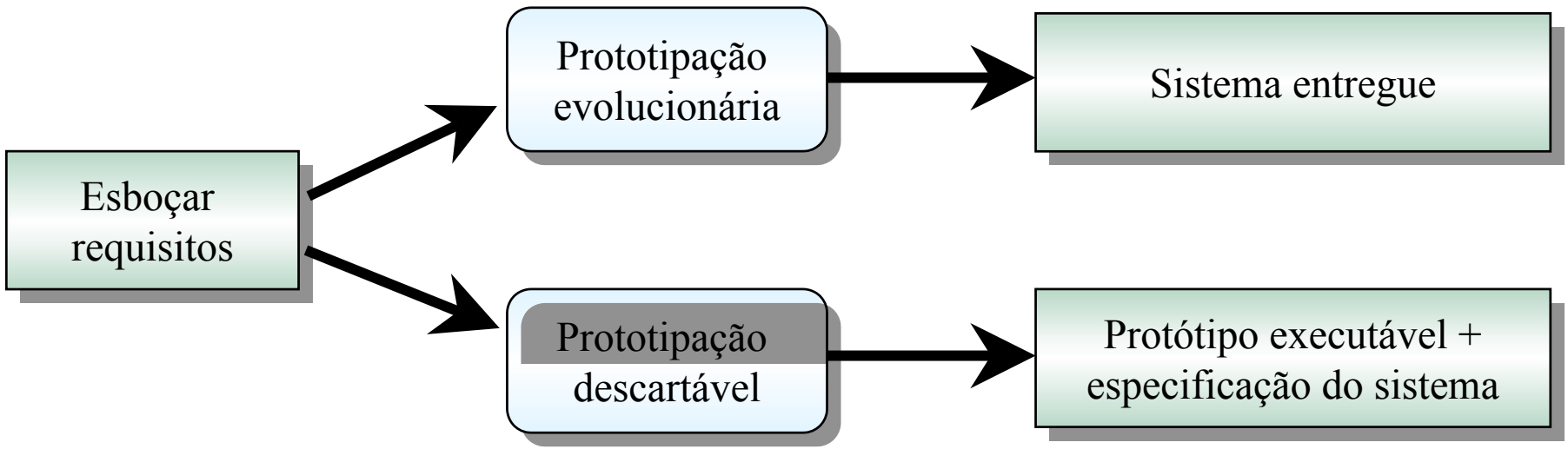
Prototipação no processo de software

- Prototipação evolucionária
  - Uma abordagem para o desenvolvimento do sistema onde um protótipo inicial é produzido e refinado através de vários estágios até atingir o sistema final.
- Prototipação descartável
  - Um protótipo o qual é usualmente uma implementação prática do sistema é produzida para ajudar a levantar os problemas com os requisitos e depois descartado. O sistema é então desenvolvido usando algum outro processo de desenvolvimento.

Objetivos da prototipação

- O objetivo da *prototipação evolucionária* é fornecer aos usuários finais um sistema funcinando. O desenvolvimento começa com aqueles requisitos que são melhores compreendidos.
- O objetivo da *prototipação descartável* é validar ou derivar os requisitos do sistema. O processo de prototipação começa com aqueles requisitos que não são bem compreendidos.

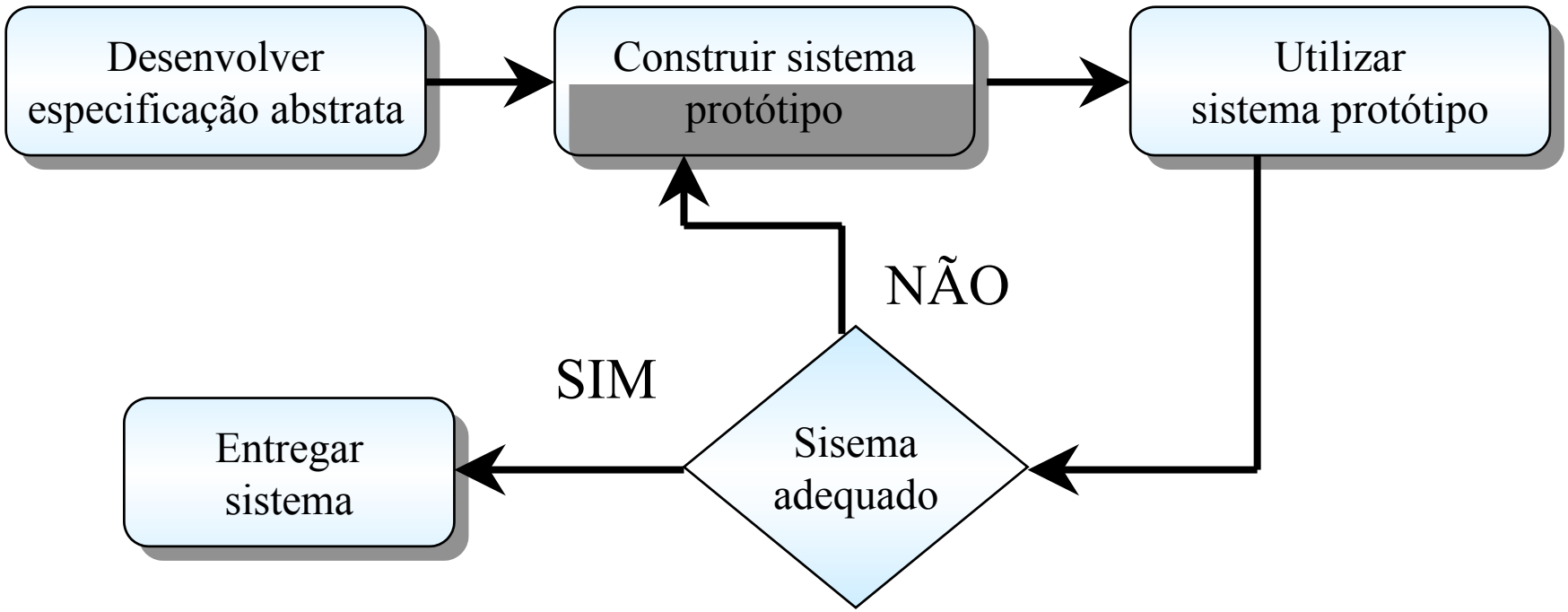
Abordagens de prototipação



Prototipação evolucionária

- Deve ser usada para sistemas onde a especificação não pode ser desenvolvida à priori, como por exemplo, os sistema de IA e os sistemas de interface com o usuário
- Baseada em técnicas que permitem interações rápidas para o desenvolvimento de aplicações.
- Verificação é impossível uma vez que não exise especificação. A validação significa demonstrar a adequação do sistema.

Prototipação evolucionária



Vantagens da prototipação evolucionária

- Rápido fornecimento do sistema
  - Em alguns casos, o rápido fornecimento e a facilidade de uso são mais importantes do que os detalhes de funcionalidade ou a facilidade de manutenção de software a longo prazo.
- Compromisso do usuário com o sistema
  - O envolvimento do usuário com o sistema significa maior possibilidade de atender aos seus requisitos e um maior empenho para que o sistema funcione de acordo.

Prototipação Evolucionária

- O processo de especificação, projeto e implementação são intercalados.
- O sistema é desenvolvido em uma série de estágios que são entregues ao cliente.
- Técnicas para o desenvolvimento rápido de sistemas, tais como ferramentas CASE e linguagens de 4a. Geração, são utilizadas.
- As interfaces com o usuário do sistema são usualmente desenvolvidas utilizando-se um sistema de desenvolvimento interativo (Lote de ferramentas GUI)

Problemas com prototipação evolucionária

- Problemas de gerenciamento
  - Processos de gerenciamento existentes assumem o modelo de desenvolvimento cascata.
  - Habilidades especialistas são necessárias e podem não estar disponível na equipe de desenvolvimento
- Problemas de manutenção
  - A continuidade de mudanças tende a corromper a estrutura do protótipo do sistema, assim a manutenção a longo prazo pode ser cara.
- Problemas contratuais
  - Os contratos são, geralmente, estabelecidos baseados em uma especificação completa do software.

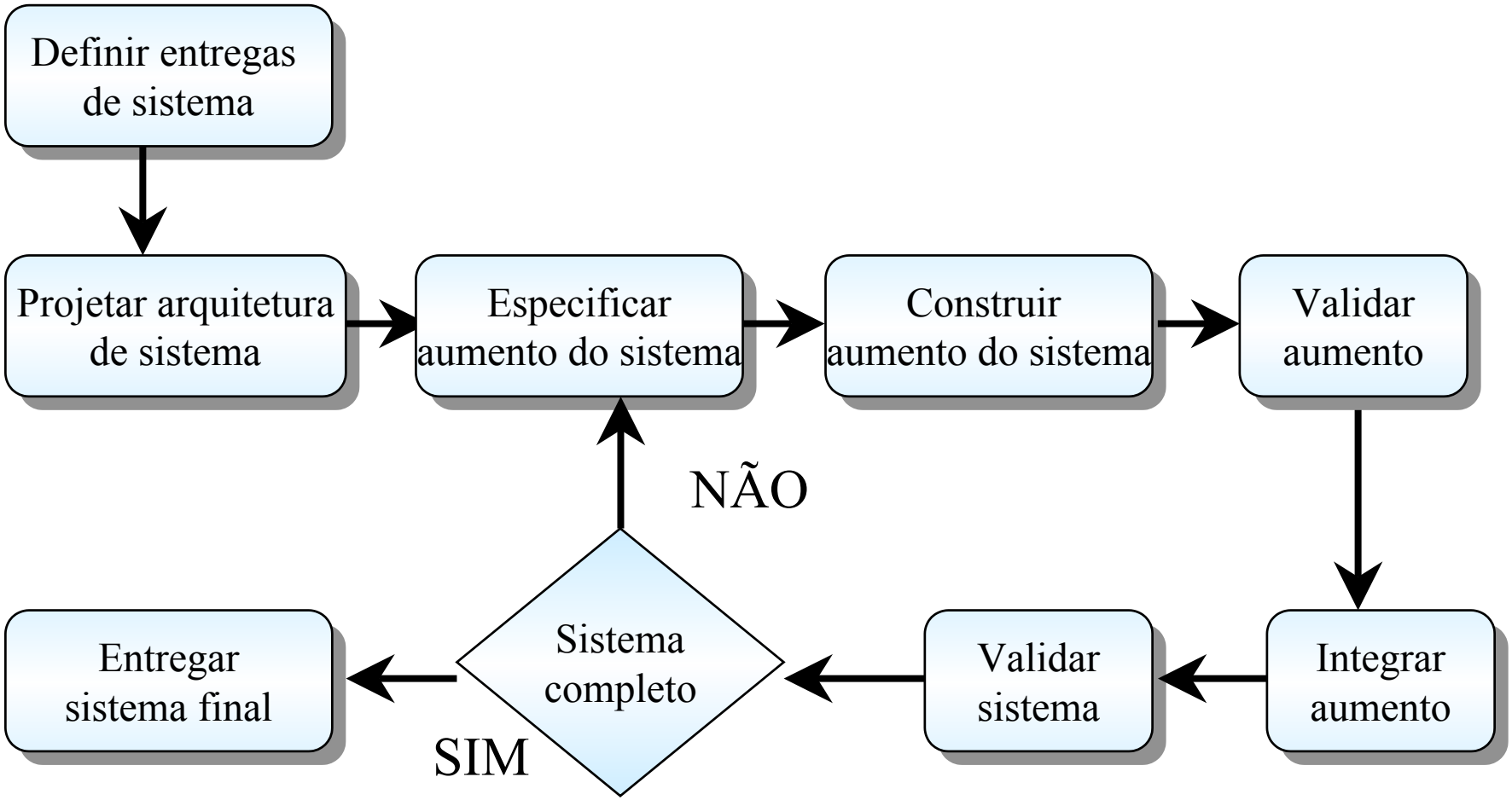
Protótipos como especificações

- Algumas partes dos requisitos (por ex. funções críticas com relação à segurança) são difíceis de aparecerem em protótipos, assim acabam não aparecendo na especificação.
- Uma implementação não tem valor legal de contrato.
- Requisitos não funcionais não podem ser testados adequadamente em um protótipo do sistema.

Desenvolvimento incremental

- O sistema é desenvolvido e liberado em incrementos após estabelecer uma arquitetura global.
- Requisitos e especificações para cada incremento podem ser desenvolvidos.
- Usuários podem avaliar os incrementos liberados enquanto outros estão sendo desenvolvidos. Portanto, esse serve como uma forma de sistema protótipo.
- O desenvolvimento incremental combina as vantagens da prototipação evolucionária com um processo de desenvolvimento mais fácil de ser gerenciado e uma melhor estruturação do sistema.

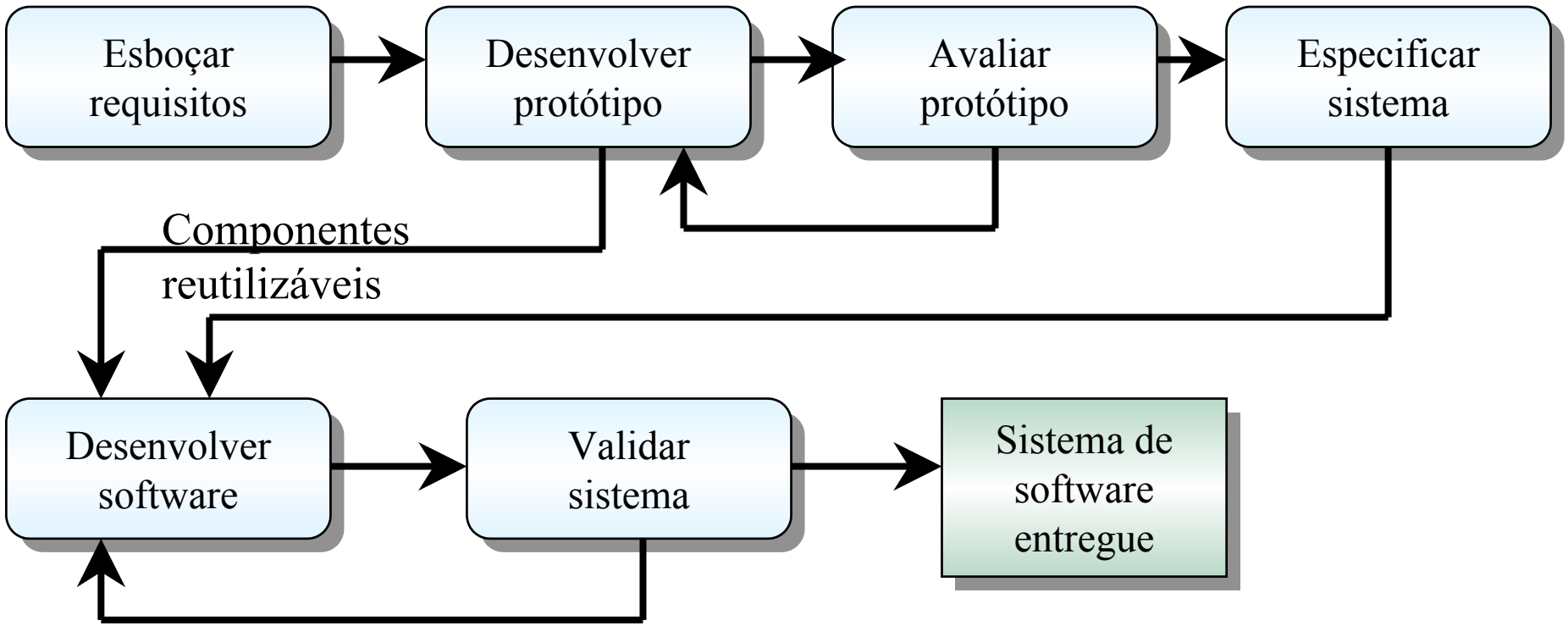
Um processo de desenvolvimento incremental



Prototipação descartável

- Usada para reduzir os riscos com os requisitos.
- O protótipo é desenvolvido de uma especificação inicial, entregue para avaliação e então descartado.
- O protótipo descartável NÃO deve ser considerado como um sistema final.
  - Características importantes podem ter sido excluídas do protótipo.
  - Não existe especificação para manutenção futura
  - O sistema será mal estruturado e difícil de manter.

Processo de software com prototipação descartável



Protótipos descartáveis liberáveis

- Desenvolvedores podem ser pressionados a entregar um protótipo descartável como um produto final
- Isso não é recomendado
  - Pode ser impossível ajustar o protótipo para atender os requisitos não funcionais.
  - O protótipo é inevitavelmente não documentado e isso é ruim para a manutenção a longo prazo.
  - A s mudanças feitas durante o desenvolvimento do protótipo provavelmente terão degradado a estrutura do sistema.
  - Os padrões de qualidade organizacional são, normalmente, deixados de lado no desenvolvimento do protótipo.

Técnicas de prototipação rápida

- Várias técnicas podem ser usadas para o desenvolvimento de protótipos
  - Desenvolvimento com linguagem dinâmica de alto nível
  - Programação de banco de dados
  - Montagem de componentes e aplicações
- Essas técnicas não são exclusivas - são muitas vezes utilizadas em conjunto.
- Programação visual é uma parte inerente da maioria dos sistemas de desenvolvimento de protótipos.

Linguagens dinâmicas de alto-nível

- São linguagens que incluem poderosos recursos de gerenciamento de dados em *run-time*.
- Necessitam de um grande sistema de suporte de run-time. Assim, não eram largamente usadas para o desenvolvimento de grandes sistemas.
- Algumas linguagens oferecem excelentes facilidades de desenvolvimento de interface com o usuário
- Algumas linguagens tem um ambiente de suporte integrado cujas faciliades podem ser usadas no protótipo.

Linguagens de alto nível para prototipação

Linguagem	*Tipo*	*Domínio de aplicação*
Smaltalk	Orientada a objetos	Sistemas interativos
Java	Orientada a objetos	Sistemas interativos
Prolog	Lógica	Processamento simbólico
LISP	Com base em listas	Processamento simbólico

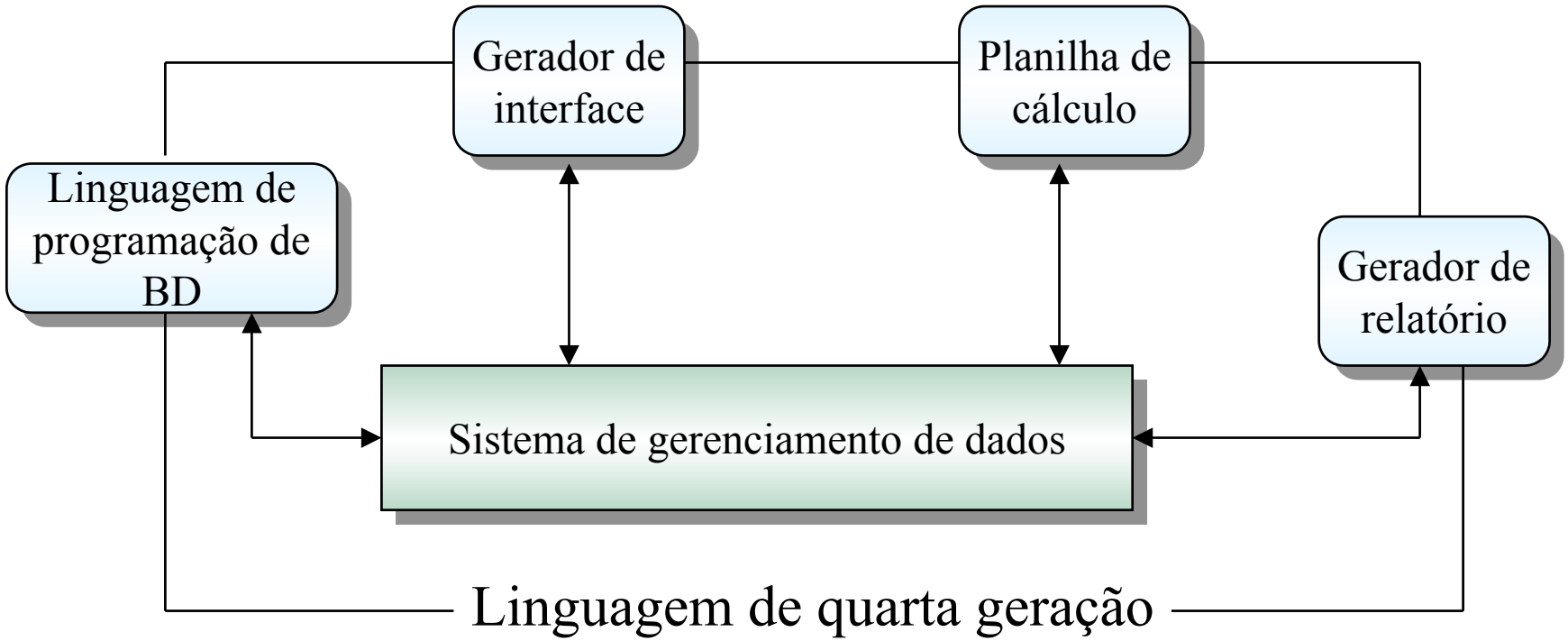
Escolha da linguagem de prototipação

- Qual é o domínio de aplicação do problema?
- Que tipo de interação com o usuário é necessário?
- Qual ambiente de suporte vem com a linguagem?
- Diferentes partes do sistema podem ser programados em diferentes linguagens. Contudo, pode haver problemas com a comunicação entre as linguagens.

Linguagens de programação de banco de dados

- Linguagens específicas ao domínio de sistemas de negócios que envolve a manipulação de dados a partir de um banco de dados.
- Normalmente inclui uma linguagem de consulta de banco de dados, um gerador de interface, um gerador de relatórios e uma planilha de cálculos.
- A linguagem + ambiente é conhecido como uma linguagem de quarta geração (4GL)
- São adequadas para sistemas de negócios de tamanho pequeno ou médio.

Componentes de linguagens de quarta geração



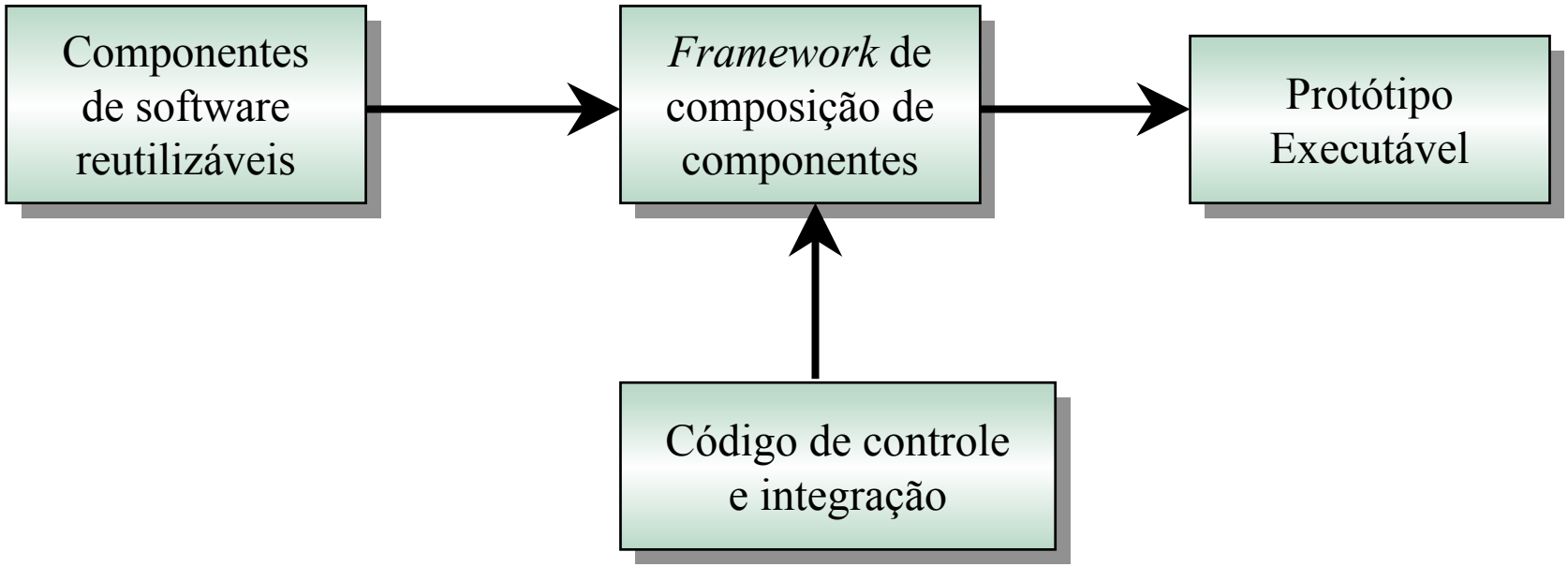
Montagem de componentes e aplicações

- Protótipos podem ser construídos rapidamente através de um conjunto de componentes reutilizáveis e um mecanismo para compor esses componentes.
- O mecanismo de composição deve incluir facilidades de controle e um mecanismo para comunicação de componentes.
- A prototipação com componentes reutilizáveis envolve desenvolver uma especificação que leva em conta a disponibilidade e funcionalidade de componentes existentes.

Prototipação com reuso

- Desenvolvimento a nível de aplicações
  - Sistemas inteiros são integrados com o protótipo , de modo que sua funcionalidade pode ser compartilhada.
  - Por exemplo, se a capacidade de edição de texto é necessária, um sistema padrão de edição de texto pode ser integrado.
- Desenvolvimento a nível de componentes
  - Componentes individuais são integrados dentro de um *framework*-padrão a fim de implementar o sistema
  - *Framework* pode ser uma linguagem de *scripting* (Visual Basic ou Perl) ou um *framework* de integração (CORBA ou JavaBeans)

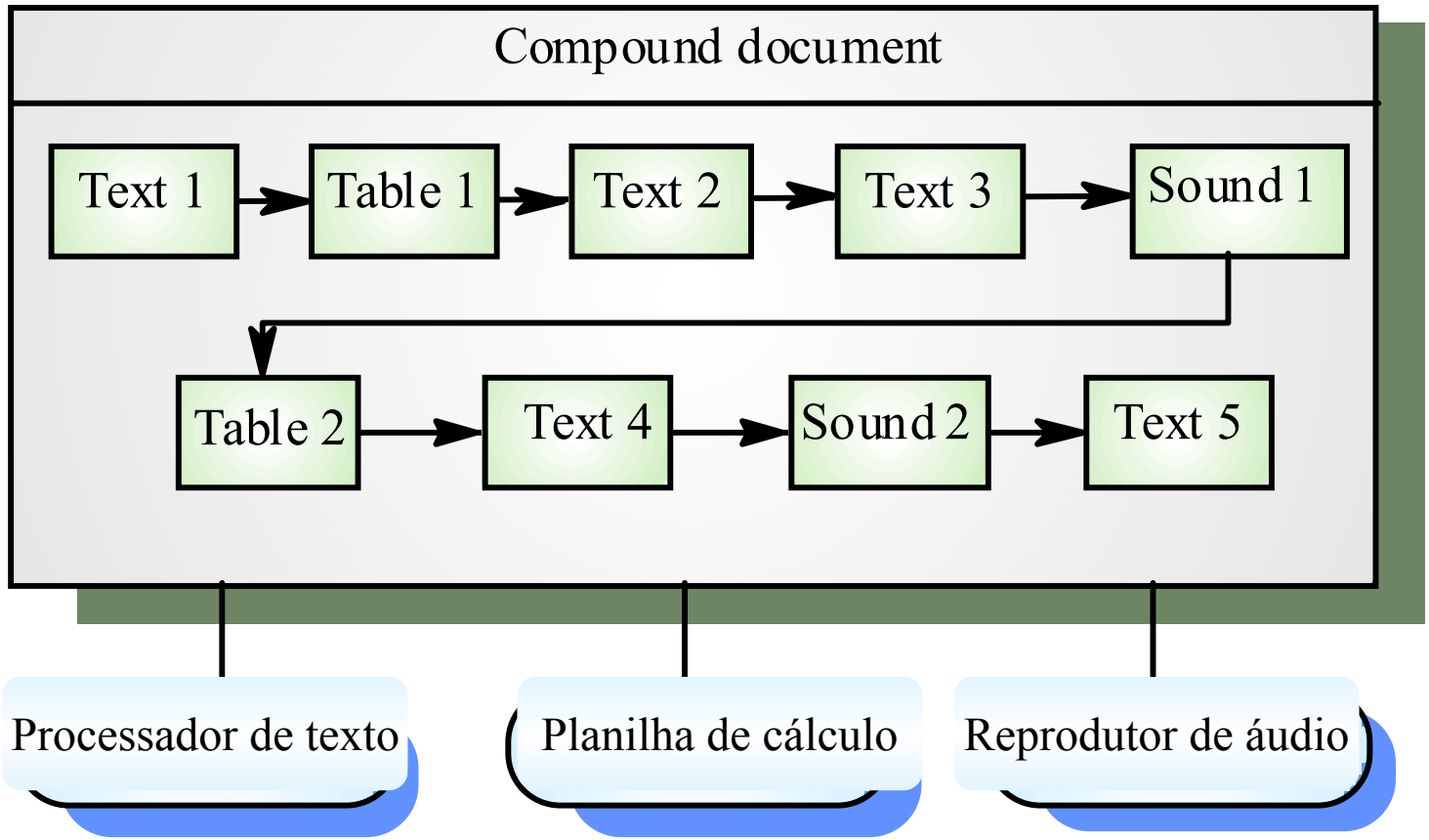
Composição de componentes reutilizáveis



Documentos compostos

- Para algumas aplicações, um protótipo pode ser criado por desenvolver um documento composto.
- Isso é um documento com elementos ativos (tal como uma planilha de cálculo) que permite funcionalidade ao usuário.
- Cada elemento ativo está associado a um aplicativo, que é chamado quando aquele elemento é selecionado.
- O próprio documento é o integrador para diferentes aplicações.

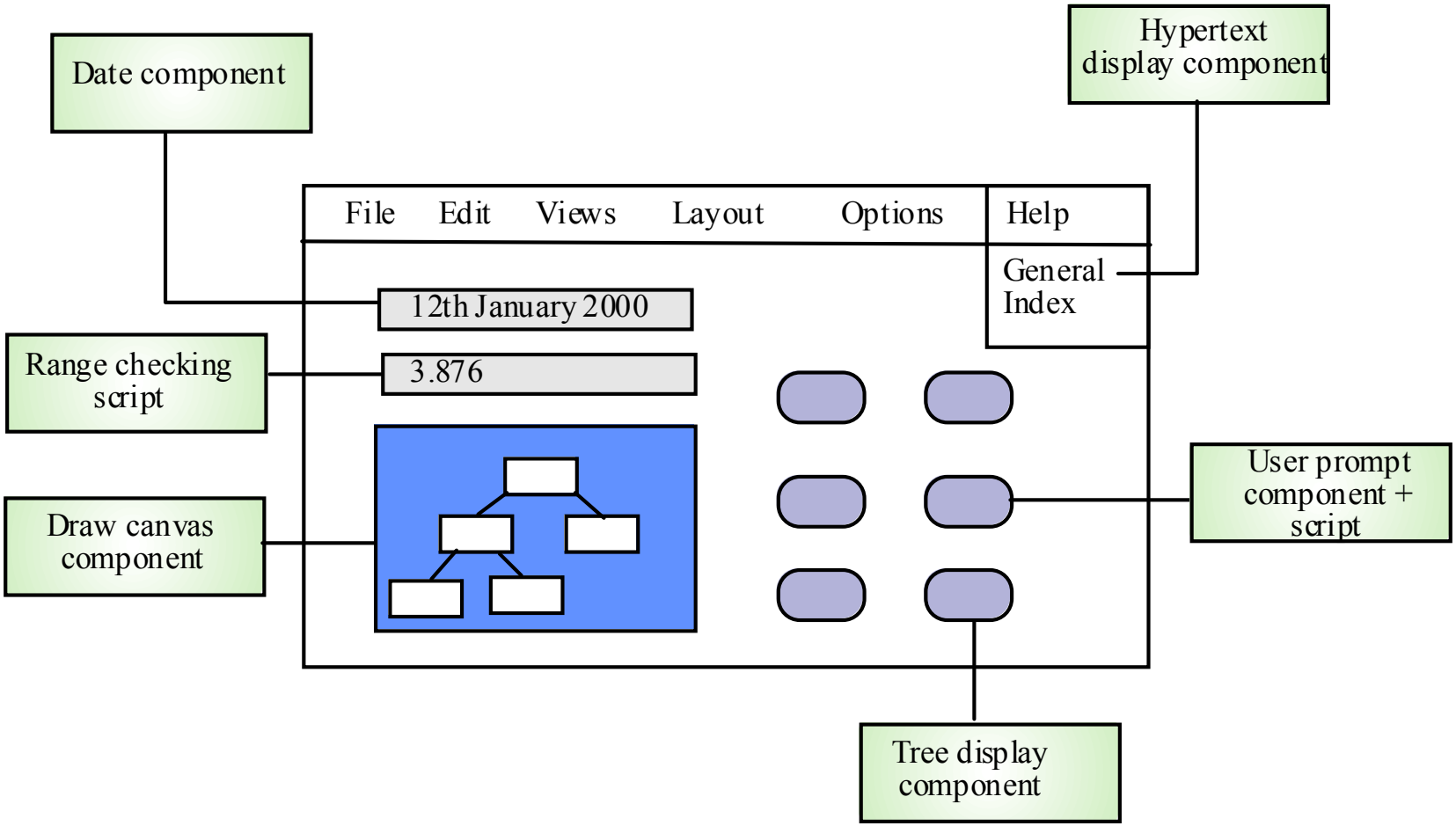
Vinculação de aplicativos.



Programação Visual com Reuso

- Linguagens de *scripting* como o *visual basic* apoiam a programação visual, onde o protótipo é desenvolvido através da criação de interface com o usuário a partir de itens padrões (telas, campos, botões e menus) e a associação de componentes à esses itens.
- Uma grande biblioteca de componentes existe para suportar esse tipo de desenvolvimento.

Programação visual com reuso



Problemas com o desenvolvimento visual

- Dificuldade de coordenar desenvolvimento em equipe.
- Não existe uma arquitetura explícita do sistema.
- Dependências complexas entre partes do programa podem causar problemas com a manutenção do sistema.

Prototipação de interface com o usuário

- Os projetistas não devem opinar a respeito de uma interface com o usuário que seja aceitável. A prototipação é essencial nesse caso.
- O desenvolvimento de IU consome uma parte substancial dos custos de desenvolvimento de aplicações.
- Os geradores de interface podem ser utilizados para projetar a interface e sua funcionalidade pode ser obtida através de componentes associados com as entidades da interface. (menus, campos, botões, etc.)
- Interfaces web podem ser prototipadas através do uso de um editor de páginas web.

Pontos-chave

- Um protótipo de sistema pode ser usado para dar aos usuários finais uma impressão concreta das capacidades desse sistema.
- A prototipação está se tornando cada vez mais comum para o desenvolvimento de sistema onde o desenvolvimento rápido é essencial.
- Protótipos descartáveis são usados para a compreensão dos requisitos do sistema.
- Na prototipação evolucionária, o sistema é desenvolvido pela evolução de uma versão inicial em uma versão final do sistema.

Pontos-chave

- O desenvolvimento rápido é importante na prototipação de sistemas. Isso pode levar à exclusão de algumas funcionalidades do sistema ou na diminuição dos requisitos não funcionais.
- Entre as técnicas de prototipação estão o uso de linguagens de nível muito elevado, a programação de bando de dados e a construção de protótipos a partir de componentes reutilizáveis.
- A prototipação é essencial para o desenvolvimento de interfaces com o usuário, as quais são difíceis de serem especificadas usando um modelo estático. Os usuários deveriam estar envolvidos na avaliação e na evolução do protótipo.


Film